



MINISTERUL EDUCAȚIEI ȘI CERCETĂRII



Informatică

Învățământ liceal

Repere metodologice
pentru consolidarea achizițiilor
anului școlar 2019-2020

Coordonator metodologic
cercet. st. dr Gabriela Nausica Noveanu – U.C.E.-C.N.P.E.E.

Autori:

Emanuela Cerchez
Daniela Lica
Carmen Popescu
Filonela Bălașa
Simona Drăghici

Colegiul Național „Emil Racoviță” Iași
Colegiul Național "Ion Luca Caragiale" Ploiești
Colegiul Național "Gheorghe Lazăr" Sibiu
Colegiul Național "Grigore Moisil", București
Colegiul Național "Ion Luca Caragiale", Ploiești

CENTRUL NAȚIONAL DE POLITICI ȘI EVALUARE ÎN EDUCAȚIE



Machetat la
Editura Didactică și Pedagogică S.A.

Cuprins

Notă de prezentare.....	4
Clasa a X-a (matematică-informatică, intensiv informatică).....	7
Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar.....	7
Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior	9
Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare ...	15
3.1. Exemple de activități remediale.....	15
3.2. Exemple de activități de recuperare	27
Clasa a XI-a (matematică-informatică, intensiv informatică).....	38
Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar.....	38
Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior	41
Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare ...	47
3.1. Exemple de activități remediale.....	47
3.2. Exemple de activități de recuperare	62
Clasa a X-a (Matematică-informatică și Științe ale naturii).....	74
Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar.....	74
Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior	76
Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare ...	81
3.1. Exemple de activități remediale.....	81
3.2. Exemple de activități de recuperare	88
Secțiunea 4 – Adaptarea la particularitățile/categoriile de elevi în risc. Exemple de activități de învățare.....	95
Clasa a XI-a (matematică-informatică).....	100
Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar.....	100
Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior	103
Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare .	107
3.1. Exemple de activități de învățare remediale.....	107
3.2. Exemple de activitate de recuperare	114
Secțiunea 4 – Adaptarea la particularitățile/categoriile de elevi în risc. Exemple de activități de învățare.....	117
ANEXE.....	126
Anexa 1.....	126
Anexa 2.....	127
Anexa 3.....	128
Anexa 4.....	129

Notă de prezentare

Anul școlar 2020-2021 vine cu oportunități și provocări unice cauzate de perturbarea procesului de predare-învățare-evaluare din primăvara anului 2020 precum și de incertitudinea privind organizarea și desfășurarea procesului educativ (față-în-față, on-line sau combinat) în anul școlar 2020-2021.

Scopul acestui ghid este de a oferi profesorilor recomandări metodologice, repere în sprijinul proiectării activităților de evaluare a achizițiilor elevilor și a activităților remediale/de recuperare.

Ghidul este structurat pe clase:

- clasa a X-a, specializarea matematică-informatică, intensiv-informatică;
- clasa a XI-a, specializarea matematică-informatică, intensiv-informatică;
- clasa a X-a, specializările matematică-informatică și științele naturii;
- clasa a XI-a, specializarea matematică-informatică.

Clasa a IX-a nu a fost inclusă în acest ghid, din cauza faptului că elevii de clasa a IX-a nu au studiat disciplina Informatică în gimnaziu. De asemenea, clasa a XII-a nu a fost inclusă în acest ghid, datorită faptului că programa clasei a XII-a pentru profilul matematică-informatică nu permite integrarea activităților remediale/de recuperare pentru competențele specifice clasei a XI-a, păstrând logica internă a programei școlare. Totuși au fost incluse exemple de activități remediale ce pot fi abordate în clasa a XII-a specializarea matematică-informatică, intensiv-informatică.

Pentru fiecare clasă, conținutul este structurat în următoarele secțiuni:

- **Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar**

Această secțiune conține recomandări din perspectiva curriculumului scris. Ele nu înseamnă o modificare a programei școlare în vigoare, ci descriu un proces pe care fiecare profesor de informatică îl poate parcurge pentru a planifica și proiecta demersul didactic astfel încât, la finalul clasei a X-a, toate competențele specifice din programele școlare aferente clasei a IX-a și clasei a X-a să fie structurate. În mod similar este abordat demersul didactic pentru clasa a XI-a.

- **Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior**

Secțiunea 2 conține modele de itemi de evaluare inițială. La începutul anului școlar este necesară realizarea unei evaluări inițiale, care să cuprindă itemi pentru toate unitățile de învățare din programa clasei anterioare, cu diferite niveluri de complexitate. În funcție de situația clasei, profesorul poate decide să aloce două-trei săptămâni pentru recapitulare/activități de predare-învățare pentru unitățile de învățare derulate exclusiv online. Itemii de evaluare pot fi proiectați pe 3 niveluri de complexitate:

- Nivelul cognitiv 1 (cunoaștere);
- Nivelul cognitiv 2 (aplicare);
- Nivelul cognitiv 3 (raționament).

Dimensiunea **Cunoaștere** (cunoștințe declarative, cunoștințe procedurale, cunoștințe contextuale) este evidențiată prin următoarele tipuri de sarcini: reamintirea informațiilor relevante, descrierea/exprimarea cu propriile cuvinte, exemplificarea, demonstrarea cunoștințelor în legătură cu utilizarea aparatelor, echipamentelor, instrumentelor;

Dimensiunea **Aplicare** (abilitatea elevului de a aplica cunoștințe și înțelegerea conceptuală manifestată în situații-problemă) este evidențiată prin următoarele tipuri de sarcini: comparare /diferențiere, relaționarea, utilizarea de modele, interpretarea, explicarea;

Dimensiunea **Raționament** (analizarea unor situații nefamiliare, a unor contexte complexe, formularea de concluzii și explicații, luarea deciziilor, transferul de cunoștințe în situații noi sau rezolvarea unor probleme ce presupun identificarea unei strategii de lucru): analiza, sinteza, formularea de întrebări/ipoteze/predicții, designul investigațiilor, evaluarea, justificarea concluziilor.

Modalitatea în care profesorul desfășoară evaluarea inițială/activitățile de învățare remediale va fi decisă de acesta în funcție de specificul clasei, asigurându-se că:

- evaluarea este corectă, transparentă și echitabilă pentru toți elevii;
- modalitatea de lucru este comunicată cu claritate elevilor și părinților la începutul anului școlar;
- activitățile de evaluare/remediale sunt proiectate astfel încât să vizeze toate competențele din programa școlară a anului școlar anterior, pe toate nivelurile;
- fiecare elev va primi un feed-back clar, constructiv, oportun și util pentru ameliorarea învățării;
- dezvoltă capacitatea elevilor de a se autoevalua, astfel încât aceștia să devină autonomi în propriul proces de învățare, prin evaluarea obiectivă a propriului progres, stabilirea obiectivelor proprii și planificarea procesului lor de învățare.

▪ **Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare**

Secțiunea 3 conține două subsecțiuni:

3.1. Exemple de activități remediale

Activitățile remediale sunt asociate itemilor de evaluare inițială. În cazul în care nu toți elevii clasei au rezolvat corect itemul corespunzător, trebuie desfășurate activități remediale în care să fie implicați toți elevii clasei sau doar cei care nu au rezolvat corect acest item, decizia aparținând profesorului, în urma analizării situației clasei.

3.2. Exemple de activități de recuperare

Activitățile de recuperare sunt activități ce pot fi integrate în planificarea activităților pentru clasa X-a, respectiv clasa a XI-a și au ca scop dezvoltarea unor competențe din programa clasei anterioare, precum și a unor competențe din programa clasei curente.

▪ **Secțiunea 4 Adaptarea la particularitățile/categoriile de elevi în risc. Exemple de activități de învățare**

Deși specializarea matematică-informatică nu se regăsește în zone defavorizate, există la acest profil elevi care provin din zone defavorizate/categoriile de risc/medii dezavantajate.

Au fost identificate următoarele grupuri de elevi:

- cu nivel educațional scăzut al părinților/lipsa interesului pentru educație din partea familiei;
- cu părinți plecați în străinătate;
- cu statut socio-economic scăzut (fără părinți, cu părinți șomeri, din familii monoparentale), cu următoarele implicații în cazul învățării online:
 - nu au/au cel mult un smartphone personal;
 - nu au/au acces restricționat la Internet;
 - folosesc același dispozitiv cu alți membri ai familiei.

Pentru clasele de matematică-informatică neintensiv, unde pot exista elevi din categoriile de risc, au fost propuse, în această secțiune, exemple de activități de învățare specifice.

Tehnologiile didactice digitale oferă instrumente prin care toate principiile enumerate în Secțiunea 2 pot fi aplicate cu mai multă ușurință. De exemplu, recomandăm profesorilor să utilizeze tehnologii:

- pentru organizarea procesului de învățare, cum ar fi: Google Classroom, Microsoft Teams, Edmodo, Moodle, etc; astfel de platforme oferă profesorilor posibilitatea de a publica materiale de tipuri diverse într-un mod sistematizat, ușor de accesat de către elevi, de a urmări activitatea fiecărui elev, de a oferi feed-back individualizat, de a urmări progresul în învățare al fiecărui elev;
- pentru comunicarea cu elevii în mod sincron (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc) sau în mod asincron (grupuri de discuții pe e-mail, pe WhatsApp, pe Facebook, etc).

Pentru sarcinile de lucru/itemii de evaluare de Nivel 1 sau 2 pot fi utilizate cu succes aplicații de testare online. De exemplu:

- pot fi create teste, utilizând Google Forms (rezultatele la test fiind importate automat pe Google Classroom), fiecare elev primind feed-back imediat în funcție de răspunsul oferit;

- pot fi create teste constituite din exerciții/aplicații simple pe platforma pbinfo.ro; exercițiile pe această platformă pot fi selectate în funcție de competențele vizate și nivelul de dificultate dorit dintr-o bază de date amplă, feed-back-ul fiind de asemenea imediat;
- dacă dorim să susținem o activitate remedială în clasă în care dorim să implicăm simultan toți elevii clasei, putem utiliza cu succes platforma Kahoot (pentru itemi cu răspuns scurt și timp redus de răspuns, această platformă creează emulație în clasă, fiind deosebit de atractivă pentru elevi).

Pentru sarcinile de lucru/itemi de evaluare de Nivel 3, recomandăm platforme pe care elevii să rezolve probleme cu evaluare online cum ar fi Pbinfo, arhiva educațională .campion, Infoarena, vianuarena, etc. Rezolvarea problemelor de pe aceste platforme necesită o analiză a problemei, elaborarea unui algoritm/unor algoritmi de rezolvare, analiza complexității algoritmilor proiectați raportat la restricțiile specificate în enunțul problemei, implementarea algoritmului într-un limbaj de programare. Codul sursă este trimis de elev pe platformă și evaluat cu ajutorul unui sistem automat de evaluare pe un set de teste calibrat astfel încât să diferențieze rezolvările în funcție de corectitudine/complexitate, iar elevul primește un feed-back imediat prin intermediul unui borderou de evaluare care conține un mesaj descriptiv pentru fiecare test de evaluare utilizat. Printr-o selecție adecvată a problemelor, în funcție de competențele vizate și de nivelul elevilor, această modalitate de lucru are o eficiență maximă, deoarece formează competențe de nivel înalt; în plus permite urmărirea progresului elevilor și individualizarea procesului de învățare. Utilizarea acestui mod de lucru dezvoltă capacități de gândire de nivel înalt și totodată formează o atitudine „profesionistă”: trebuie să „livrezi” un produs funcțional care să respecte întocmai specificațiile tehnice. Rigurozitatea este un criteriu eliminatoriu. Creativitatea face diferența între diferite abordări, concretizate în diferite punctaje.

Recomandăm de asemenea organizarea unor activități de învățare complexe, concretizate prin realizarea unei aplicații *software* în echipă. Proiectarea unor astfel de activități de învățare trebuie să formeze competențele din programa școlară și în plus să dezvolte competențe esențiale pentru profesiile secolului XXI:

- *responsabilitate*: elevul își îndeplinește sarcinile de lucru și le predă în formatul solicitat, respectând timpul limită de predare; elevul își asumă responsabilitatea pentru propriul comportament;
- *organizare*: elevul realizează un plan în care detaliază sarcinile pe care trebuie să le îndeplinească pentru a finaliza cu succes activitatea; stabilește priorități; gestionează eficient timpul de lucru; evaluează obiectiv modul în care și-a îndeplinit sarcinile de lucru; alege resursele informaționale și tehnologice adecvate pentru îndeplinirea sarcinilor de lucru;
- *independență în învățare*: elevul este capabil să își planifice, monitorizeze, evalueze, gestioneze propriul proces de învățare, pentru a atinge obiectivele; demonstrează că are inițiativă, curiozitate și interes pentru învățare; demonstrează creativitate în modul de abordare a unor probleme noi;
- *colaborare*: elevul este capabil să lucreze în echipă, asumându-și diferite roluri în cadrul echipei, realizându-și propriile sarcini de lucru, contribuind cu idei, opinii, soluții la buna desfășurare a întregului proiect, comunicând constructiv cu colegii de echipă, partajând resurse sau expertiza proprie, contribuind la rezolvarea conflictelor și la luarea deciziilor.

Toate activitățile de evaluare/de învățare propuse pot fi desfășurate atât în mediul on-line, cât și față-în-față. Contextul actual generat de epidemia COVID-19 impune abordarea procesului educațional prin soluții alternative având la bază noile tehnologii și medii virtuale de învățare. Aceasta reprezintă atât o nevoie, cât și o oportunitate.

Clasa a X-a (matematică-informatică, intensiv informatică)

Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar

În vederea elaborării planificării calendaristice se recomandă următorul demers:

- se analizează programele școlare în vigoare, clasele a IX-a și a X-a, disciplina Informatică
 - o programa clasei a IX-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_intensiv_clasa%20a%20IX-a.pdf
 - o programa clasei a X-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_intensiv_clasa%20a%20X_a.pdf
- din programa clasei a IX-a se extrag acele competențe specifice a căror structurare ar fi trebuit realizată preponderent în semestrul al II-lea (în perioada în care învățarea s-a realizat la distanță) și conținuturile aferente;
- se analizează competențele specifice din programa clasei a X-a și se stabilesc legături între acestea și competențele nestructurate/ parțial structurate în anul școlar 2019-2020;
- se analizează conținuturile neabordate/parțial abordate din clasa a IX-a și se stabilește modul de fuzionare cu elementele de conținut aferente clasei a X-a, astfel încât logica internă a domeniului să fie păstrată în condițiile structurării competențelor specifice ale clasei a X-a.

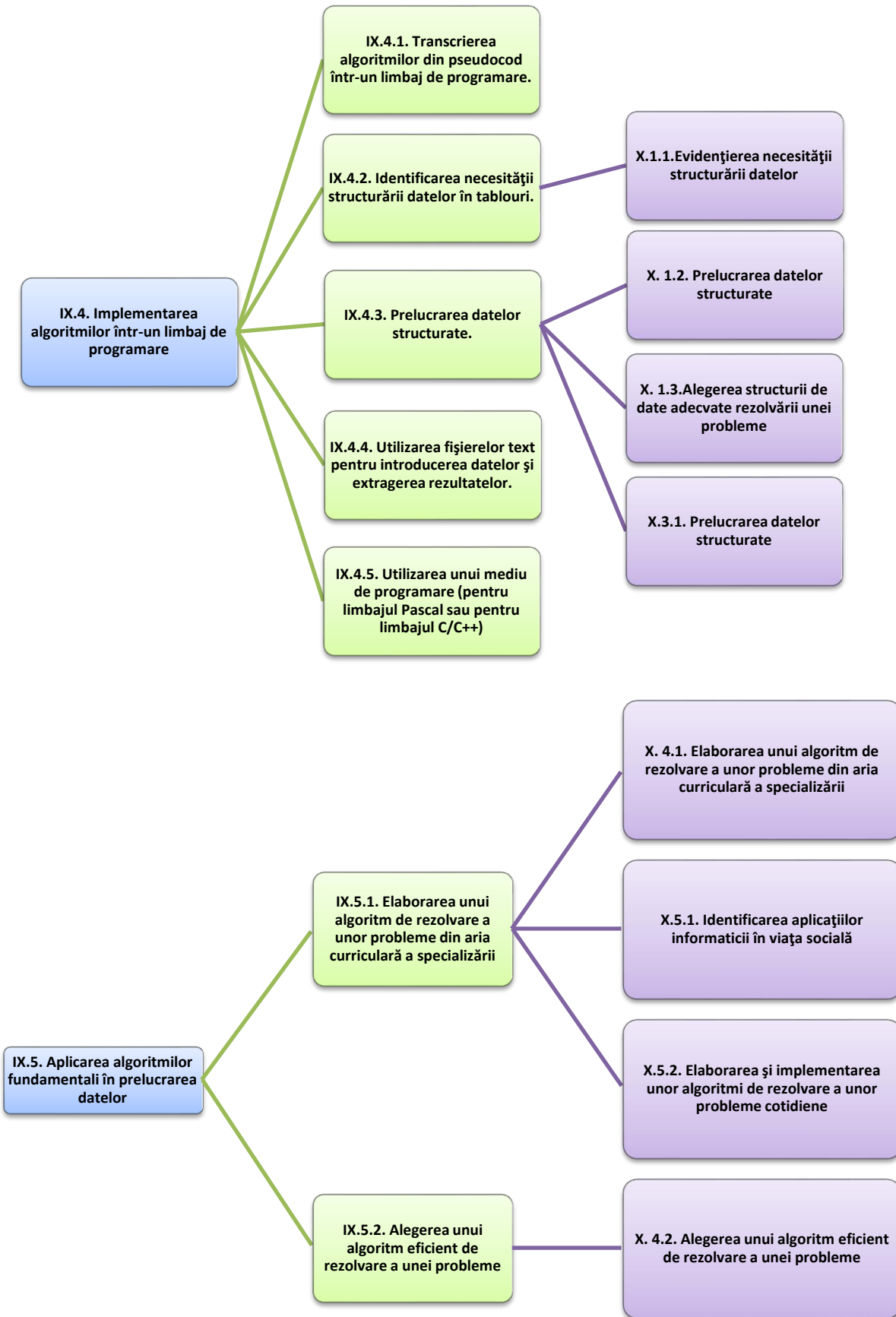
La analiza programei școlare și a planificării calendaristice a anului școlar 2019-2020, identificăm unitățile de învățare care au fost desfășurate prin activități derulate exclusiv online. Competențele specifice corespunzătoare sunt IX.4.2, IX.4.3, IX.4.5, IX.5.1, IX.5.2. În anexa 3 sunt prezentate aceste competențe și sunt reținute conținuturile, care sunt corelate cu parcursul școlar din perioada vizată de această analiză.

Competențele IX.4.2 și IX.4.3 de la clasa a IX-a, pot fi structurate, în anul școlar 2020-2021, conținuturile asociate **Tipuri structurate de date** (Tipul tablou și algoritmi fundamentali de prelucrare a datelor structurate în tablouri) vor fuziona cu conținuturile **Tipuri de date structurate** (șir de caractere, înregistrare și listă, stivă, cozi) respectiv Structuri de alocate dinamic. De exemplu, conținuturile legate de tablouri unidimensionale și bidimensionale pot fi reluate și fixate în cadrul noilor tipuri de date structurate introduse în clasa a X-a. De asemenea, în cadrul conținuturilor **Aplicații folosind subprograme** pot fi folosite exemple care impun lucrul cu tablouri.

Competența IX.4.5 va fi dezvoltată pe tot parcursul anului 2020-2021, deoarece toate conținuturile impun utilizarea unui mediu de programare.

Competențele IX.5.1 și IX.5.2 se pot structura cu ajutorul competențelor din programa clasei a X-a: X4.1, X5.1, X5.2 și X4.2 și vor fi dezvoltate în anul școlar 2020-2021, cu ajutorul conținuturilor **Aplicații din viața cotidiană**, respectiv **Aplicații interdisciplinare (specifice profilului) cu structuri de date și subprograme**.

Pentru structurarea tuturor competențelor, profesorul va încerca prioritizarea conținuturilor, dirijând învățarea în funcție de gradul de progres al elevilor. Pentru aceasta este necesară o evaluare inițială riguroasă, însoțită de activități remediale, acolo unde este necesar. În continuare este prezentată ca model o modalitate în care se poate continua dezvoltarea competențelor IX.4 și IX.5, vizate de aceste repere metodologice, de-a lungul clasei a X-a, conform programei școlare de informatică filiera teoretică, profil real, specializarea Matematică-informatică intensiv informatică și filiera vocațională, profil militar, specializarea Matematică-informatică intensiv informatică.



Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior

Item de evaluare inițială. Exemplul 1 – Nivel 1

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Enunț:

Care dintre următoarele declarații reprezintă o matrice pătratică cu 81 de elemente întregi?

- a. `int A[81][81];`
- b. `int A[81];`
- c. `int A[27][3];`
- d. `int A[9][9];`

Răspuns corect: d

Item de evaluare inițială. Exemplul 2 – Nivel 1

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Enunț:

Se consideră o matrice pătratică A cu 100 de linii și 100 de coloane declarată astfel:

```
int A[100][100];
```

Știind că atât liniile, cât și coloanele sunt numerotate începând cu 0, care dintre următoarele variante reprezintă un element situat pe diagonala secundară a matricei:

- a. `A[0][100]`
- b. `A[50][50]`
- c. `A[43][56]`
- d. `A[99][1]`

Răspuns corect: c

Item de evaluare inițială. Exemplul 3 – Nivel 2

Competența specifică:

IX.4.2 Identificarea necesității structurării datelor în tablouri

Enunț:

La un atelier lucrează doi muncitori A și B. Pentru fiecare dintre următoarele n zile ($3 \leq n \leq 10000$) se știe câte produse ar putea executa fiecare dintre cei doi muncitori: a_1, a_2, \dots, a_n , și respectiv b_1, b_2, \dots, b_n ($1 \leq a_i, b_i \leq 10000$, pentru $1 \leq i \leq n$). Atelierul fiind mic, doar un muncitor poate lucra într-o zi, iar schimbarea muncitorului de la o zi la alta nu e o bună soluție. Știind că se acceptă o singură dată schimbarea muncitorului, aflați numărul maxim de produse ce pot fi executate în cele n zile, ce muncitor trebuie să înceapă munca și după câte zile e schimbat de celălalt muncitor. Pentru rezolvarea eficientă a problemei ...

- a. este necesară utilizarea cel puțin a unui vector caracteristic
- b. este necesară utilizarea cel puțin a unui vector de sume parțiale
- c. este necesară utilizarea cel puțin a unui vector de frecvență
- d. nu este necesară utilizarea niciunui vector.

Răspuns corect: b

Item de evaluare inițială. Exemplul 4 – Nivel 2**Enunț**

Se consideră următorul algoritm, descris în pseudocod

```
int n, i, lg, lgmax;
int a[10000];
Citește n;
i=1;
Cât timp (i<=n)
    { Citește a[i];
      i=i+1;
    }
i=2; lg=2; lgmax=0;
Cât timp (i<=n)
    {i=i+1;
     Dacă (a[i]-a[i-1]==a[i-1]-a[i-2]) lg++;
     altfel
     Dacă (lg>lgmax) {lgmax=lg; lg=2;}
    }
Scrie lgmax;
```

Rezolvați, în ordine, următoarele cerințe!

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

1. Ce va afișa algoritmul dacă se citesc în ordine valorile următoare?

16 7 9 11 9 7 5 3 3 3 3 6 9 12 15 18 20

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

2. Scrieți o secvență formată din cel puțin 4 numere naturale nenule pentru care algoritmul va afișa valoarea 0.

Competența specifică:

IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare.

3. Implementați algoritmul în limbajul C⁺⁺.

Competența specifică:

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

4. Modificați programul astfel încât să determine lungimea celei mai lungi secvențe care are proprietatea de a fi progresie aritmetică.

Competența specifică:

IX.4.2 Identificarea necesității structurării datelor în tablouri

5. Este necesară utilizarea unui vector? Dacă da, justificați răspunsul! Dacă nu, modificați programul astfel încât să nu utilizeze niciun vector.

Răspuns corect:

Vezi activitatea remedială 3.1.4 „Progresie aritmetică”

EXEMPLU DE TEST DE EVALUARE ÎNȚĂLĂ

Matricea de specificații:

	Nivelul 1 (Cunoaștere)	Nivelul 2 (Aplicare)	Nivelul 3 (Raționament)
IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare	(4c)		(4d)
IX.4.2 Identificarea necesității structurării datelor în tablouri.	(1)	(5a)	
IX.4.3 Prelucrarea datelor structurate	(2), (4a)	(4b)	
IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)	(3)	(5b)	
IX.5.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării		(6a)	(5c)
IX.5.2 Alegerea unui algoritm eficient de rezolvare a unei probleme		(6c)	(6b)

Itemul 1

Competența specifică:

IX.4.2 Identificarea necesității structurării datelor în tablouri.

Se citesc de la tastatură 10 de numere naturale. Care dintre următoarele cerințe presupun realizarea unui algoritm care utilizează date structurate – tablouri unidimensionale?

- Determinați suma numerelor mai mari decât primul număr citit.
- Determinați suma dintre cel mai mic și cel mai mare număr citit
- Determinați suma numerelor mai mari decât ultimul număr citit.
- Determinați suma numerelor pare citite.

Răspuns corect: c

Itemul 2

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Considerăm un tablou bidimensional cu N linii și M coloane. Câte elemente sunt situate pe marginea tabloului (prima și ultima linie, prima și ultima coloană)?

- $2 * M + 2 * N$
- $M * N - 2 * N - 2 * M$
- $2 * N + 2 * M - 4$
- $2 * N + 2 * M - 2$

Răspuns corect: c

Itemul 3

Competența specifică:

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

Fie declarația:

Varianta Pascal	Varianta C/C++
<code>var a : array[0..2,0..2] of byte; i, j : byte;</code>	<code>int i, j, a[3][3];</code>

Specificați care vor fi elementele tabloului după execuția secvenței următoare de instrucțiuni:

<i>Varianta Pascal</i>		<i>Varianta C/C++</i>	
for i:=0 to 2 do for j:=0 to 2 do a[i,j]:=i + j;		for (i=0; i<3; i++) for (j=0; j<3; j++) a[i][j]= i + j;	
a. 1 1 1 2 2 2 3 3 3	b. 0 1 2 1 2 3 2 3 4	c. 0 1 2 0 1 2 0 1 2	d. 1 2 3 2 3 4 3 4 5

Răspuns corect: b

Itemul 4

Competențe specifice:

Item 4.a IX.4.3 Prelucrarea datelor structurate

Item 4.b IX.4.3 Prelucrarea datelor structurate

Item 4.c IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare

Item 4.d IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare

Se consideră următorul program pseudocod alăturat:

- Ce se va afișa pentru $N=6$ și tabloul unidimensional $A=\{2, 0, 1, 4, 6, 3\}$?
- Dați un exemplu de set de date de intrare pentru care nu se va efectua nici o interschimbare.
- Transcrieți în limbajul de programare studiat, algoritmul alăturat.
- Presupunem că tabloul A reține notele elevilor unei clase, la testul de informatică. Modificați algoritmul pentru a afișa pe ecran, șirul notelor ordonate descrescător.

```

citeste N;
pentru i ← 1, N executa
  citeste A[i];
pentru i ← 1, N-1 executa
  daca A[i] > A[i+1] atunci
    A[i] ↔ A[i+1];
pentru i ← 1, N executa
  scrie A[i];

```

Răspuns corect 4a: 0 1 2 4 3 6

Răspuns corect 4b: răspunsul trebuie să fie un șir de valori ordonate crescător

Răspuns corect 4c:

O implementare posibilă ar fi:

```

#include <iostream>
using namespace std;
int A[100], N;
int main() {
  cin >> N;
  for (int i=1; i<=N; i++) cin >> A[i];
  for (int i=1; i<N; i++)
    if (A[i]>A[i+1])
      {int c = A[i]; A[i]=A[i+1]; A[i+1]=c;}
  for (int i=1; i<=N; i++) cout << A[i] << " ";
  return 0;
}

```

Răspuns corect 4d:

Elevul poate alege orice metodă de sortare învățată: sortarea cu bule, prin inserție, prin selecție sau numărare. O variantă de algoritm ar putea fi:

```

citeste N;
pentru i ← 1, N executa
  citeste A[i];

```

```

pentru i ← 1, N-1 executa
  p ← i
  pentru j ← i+1, N executa
    daca A[i]>A[p] atunci
      p ← j
  A[i] ↔ A[p]
pentru i ← 1, N executa
  scrie A[i];

```

Itemul 5

Competențe specifice:

Item 5.a IX.4.2 Identificarea necesității structurării datelor în tablouri.

Item 5.b IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

Item 5.c IX.5.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Se consideră un tablou bidimensional cu N linii și M coloane, având elemente numere întregi. De pe prima linie a fișierului `test.in` se citește perechea de numere naturale N și M , de cel mult 2 cifre, iar de pe următoarele N linii, câte M numere naturale reprezentând elementele matricei.

- Scrieți o declarație de variabile necesare pentru memorarea datelor de intrare.
- Scrieți o secvență de program care permite citirea datelor de intrare.
- Scrieți o secvență de program care determină numărul de linii care au elementele dispuse în ordine strict crescătoare.

Răspuns corect 5a: `int A[100][100], N, M;`

Răspuns corect 5b:

```

ifstream fin("test.in");
fin>>N>>M;
for (int i=1; i<=N; i++)
    for (int j=1; j<=M; j++) fin>>A[i][j];

```

Răspuns corect 5c:

```

int nr = 0;
for (int i=1; i<=N; i++) {
    bool ok = true;
    for (int j=1; j<M; j++)
        if (A[i][j]>=A[i][j+1]) ok = false;
    if (ok) nr++;
}
cout<<nr<<" ";

```

Itemul 6

Competențe specifice:

Item 6.a IX.5.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Item 6.b IX.5.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Item 6.c IX.5.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Se consideră două șiruri ordonate crescător, primul format din N numere naturale, respectiv M numere naturale, toate având cel mult 9 cifre, memorate în doi vectori. Se dorește afișarea, în ordine **strict** crescătoare, a valorilor pare din cele două șiruri.

- Descrieți în limbaj natural un algoritm care folosește algoritmul de sortare pentru rezolvarea cerinței
- Scrieți un program folosind un algoritm eficient din punct de vedere al timpului de executare.
- Justificați eficiența algoritmului eficient propus.

Răspuns corect 6a

Valorile pare din cele două șiruri pot fi memorate într-un vector, care va fi sortat prin una dintre metodele învățate (selecție, inserție, metoda bulelor). După această operație, elementele de valori egale se află, plasate în vector, pe poziții alăturate. Începând cu primul element, vor fi afișate toate elementele care sunt diferite de elementul alăturat situat pe poziția precedentă ($A[i-1] \neq A[i]$).

Răspuns corect 6b

O variantă de implementare poate fi:

```
int u = 0; int i, j;
i=j=1;
while (i<=N && j<=M)
    if(A[i]%2) i++;
    else if(B[j]%2) j++;
    else if (A[i]<B[j]){
        if(A[i]>u) cout<<A[i]<<" ", u=A[i];
        i++;}
    else{
        if(B[j]>u) cout<<B[j]<<" ",u=B[j];
        j++;}
for(int t=i; t<=N; t++)
    if(A[t]>u&& A[t]%2==0)cout<<A[t]<<" ", u=A[t];
for(int t=j; t<=M; t++)
    if(B[t]>u&& B[t]%2==0)cout<<B[t]<<" ", u=B[t];
```

Răspuns corect 6c

Algoritmul propus are la bază principiul algoritmului de interclasare. Complexitatea acestuia este liniară $O(N+M)$, pe când algoritmul propus la punctul a) are o complexitate pătratică $O((N+M)^2)$, generată de algoritmul de sortare.

Timp de lucru propus: **90** minute

Notă pentru profesori:

Se va urmări ca în cadrul testului de evaluare inițială să existe minimum doi itemi pentru fiecare competență specifică evaluată.

Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare

3.1. EXEMPLE DE ACTIVITĂȚI REMEDIALE

Activitatea remedială „Declararea tablourilor bidimensionale”

Item de evaluare inițială. Exemplul 1 – Nivel 1

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea poate fi organizată în diferite moduri:

- activitate frontală, la clasă, cu implicarea preponderentă a elevilor care nu au rezolvat corect itemul de evaluare;
- activitate online – profesorul realizează un test online (utilizând Google Forms, Kahoot, Quizizz, etc) pe care îl propune spre rezolvare tuturor elevilor clasei sau doar elevilor care nu au rezolvat corect exercițiul.

Activitatea este structurată ca o conversație euristică, având scopul de a construi prin întrebări succesive capacitatea de a declara corect tablouri bidimensionale.

FIȘĂ DE LUCRU

Considerăm următoarea declarație generică a unui tablou bidimensional (matrice):

```
tip nume [LMAX] [CMAX];
```

Întrebare	Răspuns așteptat
Câte linii are tabloul declarat?	LMAX
Cum sunt numerotate liniile din tabloul declarat?	0, 1, ... LMAX-1
Câte coloane are tabloul declarat?	CMAX
Cum sunt numerotate coloanele din tabloul declarat?	0, 1, ... CMAX-1
Câte elemente are tabloul astfel declarat?	LMAX*CMAX
Care este dimensiunea zonei de memorie alocate elementelor tabloului?	LMAX*CMAX*sizeof (tip)
Revenind la exercițiul precedent, care dintre variantele a, b, c, d reprezintă un tablou cu 81 de elemente?	b, c, d
Care dintre variantele a, b, c, d reprezintă un tablou bidimensional cu 81 de elemente?	c, d
Care dintre variantele a, b, c, d reprezintă o matrice pătratică cu 81 de elemente?	d

Activitatea remedială „Diagonala secundară”

Item de evaluare inițială. Exemplul 2 – Nivel 1

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea se poate organiza în condiții similare activității „Declararea matricelor pătratice”.

FIȘĂ DE LUCRU

Pentru următoarea declarație generică a unei matrice pătratice a:

```
tip a [NMAX] [NMAX];
```

Matricea declarată are $NMAX$ linii și $NMAX$ coloane. Să notăm cu n numărul efectiv de linii/coloane ale matricei. În figura de mai jos este reprezentată o ilustrare a matricei (cu numerotarea liniilor, a coloanelor și specificarea indicilor de linie/coloană pentru fiecare element):

	0	1	2	...	$n-3$	$n-2$	$n-1$
0	$a[0][0]$	$a[0][1]$	$a[0][2]$		$a[0][n-3]$	$a[0][n-2]$	$a[0][n-1]$
1	$a[1][0]$	$a[1][1]$	$a[1][2]$		$a[1][n-3]$	$a[1][n-2]$	$a[1][n-1]$
2	$a[2][0]$	$a[2][1]$	$a[2][2]$		$a[2][n-3]$	$a[2][n-2]$	$a[2][n-1]$
...							
$n-3$	$a[n-3][0]$	$a[n-3][1]$	$a[n-3][2]$		$a[n-3][n-3]$	$a[n-3][n-2]$	$a[n-3][n-1]$
$n-2$	$a[n-2][0]$	$a[n-2][1]$	$a[n-2][2]$		$a[n-2][n-3]$	$a[n-2][n-2]$	$a[n-2][n-1]$
$n-1$	$a[n-1][0]$	$a[n-1][1]$	$a[n-1][2]$		$a[n-1][n-3]$	$a[n-1][n-2]$	$a[n-1][n-1]$

- Hașurați cu un creion colorat toate elementele situate pe diagonala secundară a matricei.
- Parcurgând diagonala secundară de sus în jos, completați în tabelul următor indicii de coloană ai elementelor parcurse:

Linie	Coloană
0	
1	
2	
...	
$n-3$	
$n-2$	
$n-1$	

Ce observați?

- Se observă că atunci când parcurgem diagonala secundară de sus în jos, indicele de linie crește din 1 în 1, iar indicele de coloană scade din 1 în 1. Formulați o relație între indicele de linie și indicele de coloană pentru elementele situate pe diagonala secundară a matricei!

Răspuns așteptat: $linie + coloană = n - 1$

- Revenind la exercițiul precedent, în care $n=100$, pentru care dintre variantele a, b, c sau d suma dintre indicele de linie și indicele de coloană este 99?

Activitate remedială „Vectori de sume parțiale”

Item de evaluare inițială. Exemplul 3 – Nivel 2

Competența specifică:

IX.4.2 Identificarea necesității structurării datelor în tablouri

Notă pentru profesori:

Propunem o activitate remedială, în două etape:

- *Etapa 1:* analiza problemei, construirea metodei de rezolvare, optimizarea soluției;
- *Etapa 2:* propunerea spre rezolvare a altor exerciții, de același tip.

FIȘĂ DE LUCRU

Etapa 1

O precizare „cheie” din enunțul problemei este că schimbarea muncitorului este permisă o singură dată. Să notăm cu d numărul de zile după care va fi schimbat muncitorul ($1 \leq d \leq n-1$).

Întrebare	Răspuns așteptat
Câte produse se execută, dacă muncitorul A începe munca?	$NrA_d = a_1 + a_2 + \dots + a_d + b_{d+1} + \dots + b_n$
Câte produse se execută, dacă muncitorul B începe munca?	$NrB_d = b_1 + b_2 + \dots + b_d + a_{d+1} + \dots + a_n$
Care variantă ar fi convenabilă?	$Nr_d = \max\{NrA_d, NrB_d\}$

Cum alegem varianta optimă global?	$\max\{Nr_d \mid 1 \leq d \leq n-1\}$
Ce complexitate ar avea algoritmul dacă la fiecare pas d ($1 \leq d \leq n-1$) în determinarea maximului am determina Nr_{A_d} , respectiv Nr_{B_d} , parcurgând elementele vectorilor a și b ?	Pentru calculul sumelor se execută n operații, iar pentru determinarea maximului $n-1$, în total obținem o complexitate $O(n^2)$.
Cum am putea optimiza algoritmul?	Pentru a nu calcula sumele în $O(n)$ la fiecare pas, vom construi, chiar de la citirea valorilor a_1, a_2, \dots, a_n , respectiv b_1, b_2, \dots, b_n doi vectori de sume parțiale: $SA_i = a_1 + a_2 + \dots + a_i, 1 \leq i \leq n$ $SB_i = b_1 + b_2 + \dots + b_i, 1 \leq i \leq n$ Calculul sumelor parțiale se poate face în $O(n)$, observând că $SA_i = SA_{i-1} + a_i$ (considerând $SA_0 = 0$) și analog $SB_i = SB_{i-1} + b_i$ Vectorii de sume parțiale fiind deja calculați, Nr_{A_d} și Nr_{B_d} ($1 \leq d \leq n-1$) se pot calcula în timp constant astfel: $Nr_{A_d} = SA_d + (SB_n - SB_d)$ $Nr_{B_d} = SB_d + (SA_n - SA_d)$ Algoritmul devine astfel liniar.

Etapa 2**Exercițiul remedial 1**

Se consideră următoarea problemă.

În perioada de vară programul angajaților unei organizații globale este mai lejer. În fiecare zi membrii organizației intră pe Internet într-un interval orar fixat de ei, același în fiecare zi. Președintele organizației ar dori să stabilească o consultare online în una dintre zile, într-un interval orar în care toți cei n ($1 \leq n \leq 100000$) membri sunt online. Cunoscând programul fiecărui membru al organizației, determinați cât timp (pe parcursul a 24 de ore) se află online simultan toți membrii. Pentru rezolvarea eficientă a problemei ...

- este necesară utilizarea cel puțin a unui vector caracteristic
- este necesară utilizarea cel puțin a unui vector de sume parțiale
- este necesară utilizarea cel puțin a unui vector de frecvență
- nu este necesară utilizarea niciunui vector.

Exercițiul remedial 2

Se consideră următoarea problemă.

Stația meteorologică "Toaca" din Masivul Ceahlău a fost de câțiva ani automatizată. Înainte, urca zilnic meteorologul de serviciu pentru a culege valorile temperaturilor de peste zi, lucru destul de anevoios deoarece traseul este dificil. Acum toate informațiile despre temperatură se centralizează automat la Bacău din 30 în 30 de minute. Un site preia un număr de nr ($2 \leq nr \leq 100000$) valori și face publică pe Internet o situație a mediei aritmetice a celor mai scăzute n valori distincte de temperatură și a mediei aritmetice a celor mai ridicate p valori distincte de temperatură dintre cele nr ($1 \leq n, p \leq nr$). Cunoscându-se n, p, nr și valorile temperaturilor, scrieți un program care calculează partea întregă a celor două medii aritmetice. Pentru rezolvarea eficientă a problemei ...

- este necesară utilizarea cel puțin a unui vector caracteristic
- este necesară utilizarea cel puțin a unui vector de sume parțiale
- este necesară utilizarea cel puțin a unui vector de frecvență
- nu este necesară utilizarea niciunui vector.

Activitatea remedială „Progresie aritmetică”

Item de evaluare inițială. Exemplul 4 – Nivel 2

Competența specifică:

IX.4.3. Prelucrarea datelor structurate

IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare.

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

IX.4.3 Prelucrarea datelor structurate

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

IX.4.2 Identificarea necesității structurării datelor în tablouri

Notă pentru profesori:

Propunem o activitate remedială, în două etape:

- Etapa 1: rezolvarea cerințelor care nu au fost abordate corect de către toți elevii clasei;
- Etapa 2: propunerea spre rezolvare a unei fișe de lucru cu un item de același tip.

FIȘĂ DE LUCRU

Etapa 1.

1. Răspuns corect: 8

Pentru a rezolva această cerință simulați execuția pas cu pas a algoritmului, urmărind într-un tabel modul în care se schimbă valorile variabilelor pe parcursul execuției algoritmului. Tabelul va conține atâtea coloane câte variabile simple există în algoritm. Vectorul citit îl veți ilustra separat cu numerotarea pozițiilor din vector și puteți începe urmărirea („watch”) după secvența de citire a vectorului.

	7	9	11	9	7	5	3	3	3	3	6	9	12	15	18	20		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...

n	i	lg	lgmax
16	2	2	0
	3	3	
	4	2	3
	5	3	
	6	4	
	7	5	
	8	2	5
	...		

2. Răspuns corect: se va introduce o secvență de valori în care primul număr (să-l notăm cu x) reprezintă numărul de valori care urmează să fie citite, apoi o succesiune de x valori care constituie o progresie aritmetică. De exemplu: 3 1 2 3

Pentru acest tip de test, în care toată secvența este o progresie aritmetică, lg_{max} va rămâne 0, deoarece lg_{max} nu va fi niciodată comparat cu lg . Lungimea secvenței curente este comparată cu lg_{max} atunci când se termină o subsecvență cu proprietatea de progresie aritmetică și începe alta nouă.

3. Implementați algoritmul în limbajul de programare studiat la clasă, folosind un mediu de programare (de exemplu, pentru limbajele C/C++ mediul CodeBlocks, pentru limbajul Pascal mediul FreePascal). O implementare posibilă:

```
#include <iostream>
using namespace std;
int main()
{int n, i, lg, lgmax;
```



```

int a[10000];
cin>>n;
i=1;
while (i<=n)
    {cin>>a[i]; i=i+1; }
i=2; lg=2; lgmax=0;
while (i<=n)
    {i=i+1;
    if (a[i]-a[i-1]==a[i-1]-a[i-2]) lg++;
    else
        if (lg>lgmax) {lgmax=lg; lg=2;}}
cout<<lgmax;
return 0;
}

```

4. Identificăm în acest program două erori (din perspectiva enunțului modificat al problemei):

- variabila `lg` care reține lungimea secvenței curente este reinițializată cu valoarea 2 dacă și numai dacă a fost identificată o subsecvență cu proprietatea de progresie aritmetică mai lungă decât cea de lungime maximă de până acum; reinițializarea variabilei `lg` trebuie realizată în orice caz, pentru că începe o nouă subsecvență pe care urmează să o analizăm;
- lungimea subsecvenței curente este comparată cu lungimea subsecvenței de lungime maximă atunci când începe o nouă subsecvență; ca urmare lungimea ultimei subsecvențe nu va mai fi comparată cu cea maximă, este necesar ca la final să adăugăm un test suplimentar.

Valorificând aceste două observații modificați programul! O abordare posibilă:

```

#include <iostream>
using namespace std;
int main()
{int n, i, lg, lgmax;
int a[10000];
cin>>n;
i=1;
while (i<=n)
    {
    cin>>a[i];
    i=i+1;
    }
i=2; lg=2; lgmax=0;
while (i<=n)
    {
    i=i+1;
    if (a[i]-a[i-1]==a[i-1]-a[i-2]) lg++;
    else
        {if (lg>lgmax) lgmax=lg;
        lg=2;}
    }
if (lg>lgmax) lgmax=lg;
cout<<lgmax;
return 0;
}

```

5. În cazul acestei probleme utilizarea unui vector nu este necesară, rezolvarea problemei se poate face chiar de la citirea valorilor. Este însă necesar să reținem la fiecare pas 3 termeni consecutivi din secvența citită, pentru a putea verifica la fiecare pas proprietatea de progresie aritmetică.

O modalitate posibilă de implementare:

```

#include <iostream>
using namespace std;
int main()
{int n, i, a, b, c, lg, lgmax;
  cin>>n>>a>>b;
  i=2; lg=2; lgmax=0;
  while (i<n)
    {
      cin>>c; i++;
      if (c-b==b-a) lg++;
      else
        {
          if (lg>lgmax) lgmax=lg;
          lg=2;}
      a=b; b=c;
    }
  if (lg>lgmax) lgmax=lg;
  cout<<lgmax;
  return 0;
}

```

Etapa 2.**FIȘĂ DE LUCRU**

Se consideră următorul algoritmul, descris în pseudocod

```

int n, i, S, Smax;
int a[10000];
Citește n;
i=0;
Cât timp (i<n)
  {
    Citește a[i];
    i=i+1;
  }
i=0; S=0; Smax=0;
Cât timp (i<n)
  {
    S=S+a[i];
    Dacă (S>Smax) Smax=S;
    i=i+1;
  }
Scrie Smax;

```

Rezolvați, în ordine, următoarele cerințe!

1. Ce va afișa algoritmul dacă se citesc în ordine valorile următoare?
9 7 0 -3 4 9 5 20 -50 30
2. Scrieți o secvență formată dintr-un număr natural nenul, urmat de cel puțin 3 numere întregi pentru care algoritmul va afișa valoarea 0.
3. Implementați algoritmul în limbajul C⁺⁺.
4. Modificați programul astfel încât să determine suma maximă a elementelor unei subsecvențe a vectorului. Subsecvența poate fi vidă.
5. Este necesară utilizarea unui vector? Dacă da, justificați răspunsul! Dacă nu, modificați programul astfel încât să nu utilizeze niciun vector.

ACTIVITĂȚI REMEDIALE CE VIZEAZĂ ASPECTELE CONSTATATE ÎN URMA APLICĂRII TESTULUI DE EVALUARE ÎNȚĂLĂ

Activitate remedială „Urna cu Bile”

Itemul 1

Competența specifică:

IX.4.2 Identificarea necesității structurării datelor în tablouri.

Notă pentru profesori:

În situația în care activitatea se desfășoară în clasă, atunci profesorul ar putea începe printr-o explicație menită să justifice alegerea răspunsului c. Profesorul poate reprezenta algoritmul corespunzător fiecărei variante de răspuns, implicând elevii care au rezolvat corect.

De asemenea, poate iniția o activitate frontală, structurată sub forma unei conversații euristice, în care acesta propune un enunț scurt, pentru ca mai apoi să solicite elevilor modificarea cerinței astfel încât să fie necesară folosirea datelor structurate de tip tablou.

Elevii trebuie încurajați să propună cât mai multe variante de cerințe a căror rezolvare necesită utilizarea datelor structurate de tip tablou unidimensional, eventual să reprezinte, pe caiete, câte un algoritm pentru una dintre ele.

Activitatea poate fi desfășurată online, folosind instrumente ce permit comunicarea sincronă, sub forma videoconferințelor (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc).

FIȘĂ DE LUCRU

Enunț	Cerința inițială	Varianta posibilă
Într-o urnă sunt introduse, succesiv, 100 de bile, fiecare dintre ele având inscripționat un număr natural.	Câte bile au avut același număr cu cel inscripționat pe prima bilă?	Câte bile au avut același număr cu cel inscripționat pe ultima bilă?

Activitate remedială „Organizarea datelor într-un tablou bidimensional”

Itemul 2

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Note pentru profesori:

Activitatea va viza înțelegerea modului de organizare a datelor într-un tablou bidimensional. Profesorul poate iniția o conversație euristică construită ca o succesiune de întrebări, ale căror răspunsuri vor ajuta elevul la înțelegerea reprezentării conceptuale a tablourilor bidimensionale.

FIȘĂ DE LUCRU

Pornind de la enunțul itemului 2, considerăm un tablou bidimensional A cu 10 linii și 15 coloane

Întrebare	Răspuns așteptat
Câte elemente sunt memorate în tabloul A?	$10 \times 15 = 150$
Câte elemente sunt situate în tabloul A pe prima linie?	15
Câte elemente sunt situate în tabloul A pe ultima coloană?	10
Câte elemente rămân în tabloul A dacă se elimină ultima linie?	$9 \times 15 = 135$
Câte elemente ale tabloului A au un număr minim de elemente vecine (alăturate pe linie sau pe coloană)?	4
Câte elemente sunt situate pe marginea tabloului A?	$10 + 10 + 15 + 15 - 4$

Activitatea poate fi desfășurată online, folosind platforme care permit realizarea testelor (Google Forms, Kahoot, Quizizz, etc).

Activitate remedială „Completarea elementelor într-un tablou bidimensional”

Itemul 3

Competențe specifice:

IX.4.3 Prelucrarea datelor structurate

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

Notă pentru profesori:

Activitatea va urmări dezvoltarea competenței de prelucrare a datelor structurate, respectiv parcurgerea și referirea elementelor unui tablou bidimensional, dar și utilizarea unui mediu de programare, la nivel de cunoaștere.

Ca sarcină de lucru, elevii vor fi sprijiniți să rezolve un item de asociere simplă. Activitatea va putea fi organizată în clasă, ca o activitate frontală, în care sunt implicați în primul rând elevii care nu au rezolvat corect itemul 3.

Întreaga activitate poate fi însă desfășurată și online, folosind instrumente de comunicare sincrone cu elevii (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc), profesorul având posibilitatea prezentării itemului și rezolvării acestuia.

FIȘĂ DE LUCRU

1. Desenați un tabel care corespunde unei matrice pătratică cu 4 linii și 4 coloane. Hașurați elementele care sunt plasate pe diagonala principală, respectiv sunt plasate pe diagonala secundară.

2. Realizați asocierile corecte între secvențele de instrucțiuni (coloana din stânga) și valorile elementelor tabloului A (coloana din dreapta), considerând următoarea declarație de variabile

```
int A[4][4], i, j;
```

1)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        A[i][j]=i;
```

2)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        A[i][j]=j;
```

3)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        if (i==j) A[i][j]=2*i;
```

4)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        if (i==j) A[i][j]=i+j;
```

5)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        if (i==3 || j==3) A[i][j]=abs(i-j);
```

6)

```
for (i=0; i<4; i++) {
    A[i][3]=abs(i-j);
    A[3][i]=abs(i-j);
}
```

7)

```
for (i=0; i<4; i++) A[i][i]= 2*i;
```

a)

0	0	0	3
0	0	0	2
0	0	0	1
	2	1	0

b)

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

c)

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

d)

8)
`for(i=0;i<4;i++) A[3][i]=A[i][3]=3-i;`

0	0	0	0
0	2	0	0
0	0	4	0
0	0	0	6

Activitate remedială „Regulă de atribuire“

Itemii 4a și 4b

Competența specifică:

IX.4.3 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a urmări modul de execuție “linie cu linie” al unui algoritm (program). Elevul va fi sprijinit să învețe parcurgerea și referirea elementelor unui tablou unidimensional.

Activitatea propusă constă în rezolvarea unei fișe de lucru ce conține itemi de completare. Activitatea va fi individuală și poate fi desfășurată în clasă, dar și online.

FIȘĂ DE LUCRU

Considerăm declarația:

```
int A[100], i;
```

1. Scrieți o expresie logică care verifică dacă indicele elementului $A[i]$ este par.
2. Completați punctele de suspensie din următoarea afirmație: Elementele situate pe poziții alăturate cu $A[i]$ sunt ...
3. Completați punctele de suspensie din secvențele următoare astfel încât elementele tabloului A să fie egale cu:

3	3	5	5	7	7	9	9	11	...	101
$A[0]$	$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$	$A[6]$	$A[7]$	$A[8]$...	$A[99]$

a.

```

pentru i ← 0, ... executa
┌
│   dacă  $i \% 2 = 0$   $A[i] \leftarrow \dots$ 
│   altfel  $A[i] \leftarrow \dots$ 
└
```

b.

```

pentru i ← 0, ..., 2 executa
┌
│    $A[i] \leftarrow \dots$ 
│    $A[i+1] \leftarrow \dots$ 
└
```

c.

```

pentru i ← 0, ... executa
┌
│    $A[i] \leftarrow \dots$ 
└
```

Activitate remedială „Joc online“

Itemul 4c

Competențe specifice:

IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a transcrie corect un algoritm în limbaj de programare, să cunoască structura, elementele de bază, instrucțiunile și sintaxa limbajului studiat.

Activitatea propusă constă în alegerea unei aplicații educaționale și parcurgerea individuală a acesteia. De exemplu, pe platforma educațională .campion se poate alege unul dintre jocurile propuse (Labirint, Rebus, Aritmogrif, sau Pescuitorul de perle), în cadrul aplicației “Elemente de bază ale limbajului C/C++” http://campion.edu.ro/arhiva/www/arhiva_2009/seds/10/index.htm

Elemente de bază ale limbajului C/C++

Index

Lecția 1

- Setul de caractere
- Identificatori
- Cuvinte rezervate
- Constantele întregi
- Constantele reale
- Constantele caracter
- Constantele șir de caractere

Lecția 2

- Codificarea numerelor naturale
- Codificarea numerelor întregi
- Codificarea caracterelor
- Codificarea numerelor reale
- Variabile

Lecția 3

- Expresii
- Evaluarea expresiilor
- Tipul expresiilor
- Operatori aritmetici
- Operatori de incrementare/decrementare

Lecția 4

- Operatori relaționali
- Operatori de egalitate
- Operatori logici globali
- Operatori logici pe biți

Lecția 5

- Operatori de atribuire
- Operatori condiționali
- Operatorul de conversie explicită de tip
- Operatorul de determinare a dimensiunii
- Operatorul virgulă

Lecția 6

- Joc didactic Labirint
- Joc didactic Rebus
- Joc didactic Aritmogrif
- Joc didactic Pescuitorul de perle

Lecția 7

- Fixarea cunoștințelor
- Evaluare finală

Periețanu Alexandru
Răducanu Dragoș

prof. Emanuela Cerchez
prof. Marinela Șerban

O aplicație gratuită, utilă pentru dezvoltarea capacității de a transcrie și de a realiza programe în C++ este aplicația *Learn C++* disponibilă pe Android și pe iOS.

Activitate remedială „Ordonare prin selecție”**Itemul 4d****Competențe specifice:**

IX.4.2 Identificarea necesității structurării datelor în tablouri.

IX.4.3 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea remedială își propune să ajute elevii să își dezvolte capacitatea de prelucrare a datelor structurate, să înțeleagă și să utilizeze corect algoritmi de sortare. Ea poate fi realizată atât frontal, cât și online. Dacă activitatea se va desfășura online, se vor putea folosi instrumentele sincrone de comunicare cu elevii.

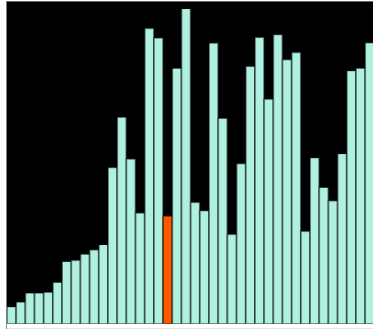
FIȘĂ DE LUCRU

Urmăriți exemplul și animația care descrie pașii algoritmului de sortare prin selecție disponibile la adresa <https://www.pbinfo.ro/articole/5605/sortarea-prin-selecție#intlink-3>

Algoritmul descris mai sus se mai numește **sortare prin selecție generală**, sau **implicită**. O altă variantă este următoarea, în care pentru fiecare secvență $i \dots n-1$ se determină explicit minimul și se interschimbă cu $X[i]$.

```
int n, X[100];
//citire X[] cu n elemente
for(int i = 0 ; i < n - 1 ; i ++ )
{
    int p = i;
    for(int j = i + 1 ; j < n ; j ++ )
        if(X[j] < X[p])
            p = j;
    int aux = X[i];
    X[i] = X[p];
    X[p] = aux;
}
```

Animație



[Vezi animația aici!](#)

Resurse online

[Selection Sort pe Wikipedia](#)

[Animație pas cu pas](#)

[Animație, valori configurabile](#)

După vizionarea animației, rezolvați următoarele sarcini de lucru:

1. Considerăm un tablou unidimensional $X = \{3, 1, 7, 4, 9\}$. Scrieți pe caiete, care sunt elementele vectorului după fiecare etapă de execuție a algoritmului de selecție.
2. Modificați codul sursă al sortării prin selecție, astfel încât să fie ordonate descrescător doar valorile pare ale tabloului X .

Activitate remedială „MaxMat”

Item 5

Competențe specifice:

IX.4.3 Prelucrarea datelor structurate

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

IX.5.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Notă pentru profesori:

Activitatea propusă constă în două etape:

1. Etapa în care sunt dezvoltate competențele de prelucrare a datelor structurate și utilizarea unui mediu de programare la nivel 1 (cunoaștere). Elevul va recapitula operațiile de parcurgere și de referire la elementele unui tablou bidimensional.
2. Etapa constă într-un exercițiu remedial de același tip.

Etapa 1 – recapitulare

Propunem activitățile remediale ”Organizarea datelor într-un tablou bidimensional” și/sau ”Completarea elementelor într-un tablou bidimensional”. Ambele activități pot fi realizate frontal, în clasă/laborator, dar și online.

Etapa 2 – exercițiul remedial de același tip

Enunț

Se dă o matrice cu N linii și M coloane și elemente numere întregi.

Cerință

Să se determine, pentru fiecare linie a matricei, valoarea maximă și indicele coloanei pe care se află valoarea maximă.

Date de intrare

Fișierul de intrare `maxmat.in` conține pe prima linie numerele N și M , iar următoarele N linii câte M numere întregi, reprezentând elementele matricei.

Date de ieșire

Fișierul de ieșire `maxmat.out` va conține N linii; pe fiecare linie se vor afla două numere, separate prin spațiu, reprezentând valoarea maximă de pe linia curentă și indicele coloanei pe care se află aceasta.

Restricții și precizări

- $1 \leq N, M \leq 25$
- elementele matricei aparțin intervalului $[-1000, 1000]$
- dacă pe o linie a matricei, elementul maxim apare de mai multe ori, se va afișa indicele de coloană minim
- liniile și coloanele matricei sunt numerotate de la 1

(<https://www.pbinfo.ro/probleme/191/maxmat>)

Activitate remedială „Interclasări”**Item 6****Competențe specifice:**

IX.4.3 Prelucrarea datelor structurate

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

IX.5.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

IX.5.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Notă pentru profesori:

Activitatea propusă constă în două etape:

1. Etapa premergătoare constă în reactualizarea cunoștințelor referitoare la o operație de prelucrare a datelor structurale de tip tablou unidimensional – interclasarea;
2. Exercițiul remedial de același tip

FIȘA DE LUCRU**Etapa 1**

1. Vizionați animația ce ilustrează pașii algoritmului de interclasare disponibil la adresa <https://www.pbinfo.ro/articole/5588/interclasarea-tablourilor>
2. Descrieți, în pseudocod, algoritmul de interclasare.

Etapa 2 – exercițiul remedial de același tip**Enunț**

Se dau două șiruri A și B , cu N , respectiv M elemente, numere naturale, ordonate strict crescător. Să se afișeze, în ordine strict crescătoare, valorile existente în ambele șiruri.

Date de intrare

Fișierul de intrare `interclasare2.in` conține pe prima linie numărul N ; urmează N numere naturale, ordonate strict crescător, ce pot fi dispuse pe mai multe linii. Linia următoare conține numărul M și urmează M numere naturale, ordonate strict crescător, ce pot fi dispuse pe mai multe linii.

Date de ieșire

Fișierul de ieșire `interclasare2.out` va conține, în ordine strict crescătoare, valorile existente în ambele șiruri. Aceste valori vor fi afișate câte 10 pe o linie, separate prin spații. Ultima linie poate conține mai puțin de 10 de valori.

Restricții și precizări

$1 \leq N, M \leq 100000$

3.2. EXEMPLE DE ACTIVITĂȚI DE RECUPERARE

1. Activitatea de învățare „Frații”

Competențe specifice

IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare

IX.4.2 Identificarea necesității structurării datelor în tablouri

IX.4.3/X.3.1 Prelucrarea datelor structurate

IX.4.4 Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

IX.5.1/X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

IX.5.2/X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

X.2.1 Utilizarea corectă a subprogramelelor predefinite și a celor definite de utilizator

X.2.2 Construirea unor subprograme pentru rezolvarea subproblemelor unei probleme

X.3.2 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme

Notă pentru profesori:

Recomandăm dezvoltarea competențelor specifice prin aplicații de sinteză pentru unitatea de învățare „Subprograme” și „Tipuri structurate de date” din programa clasei a X-a.

Activitatea are mai multe sarcini de lucru, gradate ca nivel de dificultate: primele două sarcini de lucru sunt de nivel 1, următoarele două sunt de nivel 2, iar ultima sarcină de lucru este de nivel 3.

Modul de organizare a activității la clasă este la decizia profesorului, în funcție de specificul clasei. O posibilitate de abordare ar fi ca primele 3 sarcini de lucru să fie rezolvate în clasă, individual de către fiecare elev, timpul de lucru fiind specificat de către profesor; la expirarea timpului de lucru (sau mai repede dacă toți elevii au terminat), un elev prezintă modul de rezolvare; elevii sunt solicitați să exprime opinii referitoare la corectitudinea/eficiența rezolvării și să prezinte și alte abordări (dacă există). Profesorul poate prezenta la final o altă modalitate de implementare, dacă soluțiile elevilor nu au fost optime. La finalul acestei etape a activității toți elevii clasei vor avea implementate versiuni corecte și eficiente pentru subprogramele solicitate (cu sprijinul profesorului, dacă este necesar).

A doua etapă constă în dezvoltarea aplicației educaționale. Aceasta poate fi realizată de către elevi în perechi ca sarcină de lucru pentru acasă. În funcție de nivelul clasei, profesorul poate decide să lase proiectarea aplicației în sarcina elevilor sau să prezinte o aplicație demonstrativă de acest tip, evidențiind modul de organizare a aplicației și interacțiunea aplicație-elev.

Se recomandă ca elevii să transmită aplicațiile realizate către profesor (recomandabil pe platforma utilizată pentru organizarea activității la clasă sau prin e-mail). Profesorul testează și analizează fiecare aplicație și oferă feedback elevilor, oferind acestora posibilitatea de a trimite o nouă variantă, îmbunătățită, care să valorifice feedback-ul oferit de către profesor. La clasă fiecare pereche va prezenta aplicația creată, prezentările fiind urmate de o scurtă sesiune de întrebări, opinii, aprecieri ale colegilor referitoare la aplicația prezentată.

FIȘĂ DE LUCRU

1. Efectuați pe caiet următoarele operații și transformați rezultatul în fracție ireductibilă:

$$\frac{2}{14} + \frac{5}{21}$$

$$\frac{1}{20} - \frac{13}{22}$$

$$\frac{15}{20} \cdot \frac{2}{5}$$

$$\frac{65}{12} \cdot \frac{13}{22}$$

Exprimați în limbaj natural procedeul aplicat pentru efectuarea operației. Identificați secvența de pași executată pentru efectuarea operației.

2. Proiectați o structură de date capabilă să rețină numitorul și numărătorul unei fracții (numere întregi $< 10^9$). Declarați tipul de date corespunzător.

3. Definiți următoarele subprograme:

a. subprogramul `simplifica` care primește ca parametru o fracție `f` și transformă fracția în fracție ireductibilă;

- b. subprogramul `adun` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 suma fracțiilor f_1 și f_2 în formă ireductibilă;
- c. subprogramul `scad` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 rezultatul scăderii fracției f_2 din fracția f_1 în formă ireductibilă;
- d. subprogramul `produs` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 produsul fracțiilor f_1 și f_2 în formă ireductibilă;
- e. subprogramul `impart` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 rezultatul împărțirii fracției f_1 la fracția f_2 în formă ireductibilă;
- f. subprogramul `compar` care primește ca parametri fracțiile f_1 și f_2 și returnează ca rezultat valoarea $-1, 0$ sau 1 după cum fracția f_1 este mai mică, egală, respectiv mai mare decât fracția f_2 ;
- g. subprogramul `sortare` care primește ca parametri un număr natural n și un vector F cu n fracții, memorate pe pozițiile $0, 1, \dots, n-1$ și sortează crescător fracțiile din vectorul F ;
- h. subprogramul `scrie` care primește ca parametru o fracție f și o afișează pe ecran în forma numitor/numărător.
4. Proiectați o aplicație educațională, utilă elevilor din clasele gimnaziale care studiază la matematică capitolul „Fracții”. Aplicația va avea un meniu din care elevul să poată selecta operația pe care dorește să o exerseze. Frațiile asupra cărora se aplică operația vor fi generate aleator sau selectate dintr-o listă de perechi de fracții, memorată într-un fișier text. Aplicația va utiliza subprogramele definite la punctul 3. După ce elevii rezolvă exercițiul propus (efectuează operația cu fracții) oferiți-le feedback și posibilitatea de a relua un exercițiu greșit, eventual adăugând o indicație care să îi ajute să rezolve corect. Aplicația trebuie să permită elevilor să rezolve numeroase exerciții.
5. Adăugați în meniul aplicației de la punctul 4 o opțiune care să permită elevilor avansați să exerseze operații cu fracții în expresii. Proiectați o interfață simplă, intuitivă și atractivă pentru elevi.

Evaluarea activității

Ponderea punctajelor în evaluarea activității

Cerința	1	2	3	4+5
Pondere	5%	5%	8*5%=40%	50%

Pentru fiecare criteriu sunt prezentați descriptorii pentru 4 niveluri de performanță, modul de alocare a punctajelor fiind sugerat în paranteză, dar rămânând la decizia profesorului.

Cerința 1:

Criteriu	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Rezolvarea operațiilor	Rezultatul este corect pentru fiecare operație. Modul de rezolvare ilustrează pașii urmăți.	Rezultatul este corect pentru fiecare operație. Nu este clar ilustrat modul în care a fost obținut rezultatul respectiv.	Cel puțin 3 dintre operațiile specificate au rezultat corect.	Operațiile nu sunt executate corect.
2. Exprimarea în limbaj natural cu evidențierea secvenței de pași.	Exprimarea este fluentă și corectă. Elevul se exprimă cu ușurință în limbaj științific. Pașii urmăți pentru efectuarea fiecărei operații sunt clar specificați, secvențierea fiind corectă.	Exprimarea este corectă, dar elevul are dificultăți de a transpune cu claritate în cuvinte procedeul aplicat, cel puțin pentru una dintre operații. Pașii urmăți sunt specificați cu claritate, poate nu în ordinea strictă a executării lor.	Exprimarea este sumară, termenii utilizați nu sunt științifici, pașii nu sunt delimitați cu claritate, nici secvențierea nu este corectă pentru cel puțin două dintre operații.	Elevul nu poate descrie în cuvinte procedeul folosit pentru cel puțin 3 dintre operații. Nu identifică pașii care trebuie urmați pentru efectuarea operațiilor.

Cerința 2

Criteria	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Proiectarea structurii de date	Elevul identifică necesitatea utilizării unei structuri de date care să rețină atât numitorul, cât și numărătorul fracției. Alege tipul struct/record drept cea mai bună modalitate de implementare a acestei structuri. Declară corect din punct de vedere sintactic tipul de date corespunzător.	Elevul identifică necesitatea utilizării unei structuri de date care să rețină atât numitorul, cât și numărătorul fracției. Alege tipul struct/record drept cea mai bună modalitate de implementare a acestei structuri. Are dificultăți în a declara corect din punct de vedere sintactic tipul de date corespunzător.	Elevul identifică necesitatea utilizării unei structuri de date care să rețină atât numitorul, cât și numărătorul fracției. Nu identifică tipul struct/record drept varianta optimă de implementare. Declară corect din punct de vedere sintactic o altă modalitate de implementare.	Elevul nu identifică necesitatea utilizării unei structuri de date care să rețină atât numitorul, cât și numărătorul fracției. Declarațiile sunt pentru variabile independente pentru numitor/numărător.

Cerința 3

Pentru fiecare dintre funcții se aplică următoarele criterii:

Criteria	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Antetul funcției	Antetul funcției este corect din punct de vedere sintactic. Funcția are parametri specificați. Declararea parametrilor permite transferul rezultatului calculat de funcție.	Antetul funcției este corect din punct de vedere sintactic. Funcția are parametri specificați. Declararea parametrilor nu permite transferul rezultatului calculat de funcție.	Antetul funcției este corect din punct de vedere sintactic. Funcția nu are parametri specificați. Declararea parametrilor nu permite transferul rezultatului calculat de funcție.	Antetul funcției nu este corect.
2. Implementarea operației	Subprogramul implementează corect operația. Rezultatul este transferat corect, prin intermediul parametrului.	Subprogramul implementează corect operația. Rezultatul nu este transferat corect, prin intermediul parametrului specificat.	Subprogramul implementează principial corect operația, dar nu funcționează pentru orice set de date.	Subprogramul nu implementează corect operația specificată.

Criterii de evaluare a aplicației

Criteria	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Funcționalitate	Toate elementele aplicației sunt funcționale	Majoritatea elementelor aplicației sunt funcționale	În general elementele aplicației sunt funcționale	Multe dintre elementele aplicației nu sunt funcționale
2. Organizarea aplicației	Aplicația are meniu din care pot fi selectate operațiile ce urmează a fi exersate. Aplicația are structură modulară, fiind apelate pentru fiecare operație subprogramele definite la cerința 3.	Aplicația are meniu din care pot fi selectate operațiile ce urmează a fi exersate. Aplicația este parțial modularizată.	Aplicația are meniu din care pot fi selectate operațiile ce urmează a fi exersate, dar este insuficient modularizată.	Aplicația nu are un meniu, operațiile pot fi executate secvențial.

3. Reluare, feed-back, sprijin în învățare	Aplicația permite ca o operație să fie reluată de oricâte ori, pe numeroase exemple. Elevul primește feed-back clar după rezolvarea fiecărui exercițiu. Elevul are posibilitatea de a relua un exercițiu dacă l-a rezolvat greșit. La reluarea exercițiului, elevului i se oferă informații suplimentare, care să îl ajute în rezolvarea corectă.	Aplicația permite ca o operație să fie reluată de oricâte ori, pe numeroase exemple. Elevul primește feed-back clar după rezolvarea fiecărui exercițiu. Elevul nu are posibilitatea de a relua un exercițiu dacă l-a rezolvat greșit.	Aplicația permite ca o operație să fie reluată de oricâte ori, pe numeroase exemple. Feed-back-ul este sumar, nu există posibilitate de reluare a unui exercițiu greșit.	Aplicația permite executarea unei operații pe un singur exemplu sau pe un număr foarte redus de exemple. Feed-back-ul este sumar, nu există posibilități de reluare.
4. Complexitate tehnică	Aplicația este complexă, necesită abilități de analiză și programare de nivel relativ înalt	Aplicația are complexitate medie din punctul de vedere al analizei, dar necesită bune abilități de programare	Aplicația are complexitate medie	Aplicația este simplă de analizat și implementat
5. Interfața aplicației	Interfața aplicației este funcțională, ergonomică, predictibilă și atractivă	Interfața aplicației este funcțională, intuitivă, dar ergonomia este redusă	Interfața aplicației este funcțională, dar dificil de utilizat.	Interfața este nefuncțională, neatractivă

Considerații finale

Activitatea poate fi reluată în clasa a XII-a, fiind o activitate de învățare utilă pentru unitatea de învățare “Programarea orientată pe obiecte” din programa clasei a XI-a ce poate fi integrată cu succes în programa clasei a XII-a, în cadrul unității de învățare “Programare vizuală”. Cerințele pot fi adaptate: se poate implementa o clasă denumită *fracție*; operațiile cu fracții pot fi implementate prin supraîncărcarea operatorilor; se solicită realizarea unei interfețe grafice, într-un mediu de programare vizual. Astfel de activități interdisciplinare pot fi proiectate (la nivelul clasei a X-a sau la nivelul clasei a XII-a) pentru numeroase teme: numere complexe, numere mari, polinoame, etc.

2. Activitatea de învățare „Creatorul de jocuri” – Nivel 3

Competențe specifice

IX.4.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare.

IX.4.2 Identificarea necesității structurării datelor în tablouri.

IX.4.3/X.3.1 Prelucrarea datelor structurate

IX.4.4 Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

IX.5.1/X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

IX.5.2/X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

X.2.1 Utilizarea corectă a subprogramelelor predefinite și a celor definite de utilizator

X.2.2 Construirea unor subprograme pentru rezolvarea subproblemelor unei probleme

X.3.2 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia

FIȘĂ DE LUCRU

Veți concepe și implementa un joc. Acesta va fi implementat într-un limbaj de programare și va utiliza biblioteci de grafică la alegerea voastră.

Etape

Nr	Activitate	Data limită
1.	Prezentarea activității	
2.	Formarea echipelor (2-4 membri), stabilirea coordonatorului și a denumirii echipei	
3.	Stabilirea temelor de proiect (denumire și strategie joc)	
4.	Realizarea și analizarea fișelor de proiect. Fișa* va fi publicată pe Google Drive și va fi partajată între membrii echipei și prof. coordonator	
5.	Realizarea proiectelor. Liderul echipei publică pe Google Classroom o arhivă zip care conține proiectul echipei. Arhiva va fi denumită cu numele echipei.	
7.	Reflecție asupra propriei învățări. Elevii vor consemna în jurnalul proiectului modul în care realizarea acestui proiect au contribuit la îmbunătățirea performanțelor lor.	
8.	Prezentarea proiectelor. Timp de prezentare: 10 minute; 5 minute întrebări/interevaluare	
9.	Evaluarea proiectelor. Proiectele vor fi evaluate atât de profesori, cât și de fiecare echipă de elevi. De asemenea, fiecare echipă își va autoevalua proiectul.	
10	Notarea elevilor	

*În fișa de proiect elevii vor consemna:

Nr	Activitate	Responsabil	Data	Observații

Evaluarea proiectului

Evaluarea proiectelor va fi realizată de:

- a. profesorul de informatică
- b. elevii clasei

În evaluarea proiectului se vor analiza următoarele aspecte:

- A. Realizarea proiectului (50%)
- B. Prezentarea proiectului (30%)
- C. Evaluarea și autoevaluarea realizată de elevi (10%)
- D. Consemnarea în blog-ul personal (10%)

A. Realizarea proiectului

Criteria	10-9	8-7	6-5	4-3-2-1
1. Documentație	Jocul este însoțit de o documentație externă care conține descrierea jocului și modul de utilizare a programului. Utilizatorul jocului poate consulta oricând documentația internă (help), care conține instrucțiuni de joc	Jocul este însoțit de o documentație externă / internă dar acestea sunt incomplete (nu oferă utilizatorului toate informațiile necesare)	Documentațiile sunt sumare	Jocul nu este însoțit de o documentație externă și nici internă.
2. Complexitate tehnică	Jocul este complex, necesită abilități de analiză și programare de nivel relativ înalt	Jocul are complexitate medie din punctul de vedere al analizei, dar necesită bune abilități de programare	Jocul are complexitate medie	Jocul este simplu de analizat și implementat
3. Creativitate, utilitate	Jocul este original, implică strategii de joc, este educativ	Jocul este relativ original, strategiile de sunt simple, dar	Jocul nu este original, strategiile de joc sunt	Jocul nu este original, nu are caracter educativ

		are caracter educativ	simple; are caracter educativ	
4. Funcționalitate	Toate elementele proiectului sunt funcționale	Majoritatea elementelor proiectului sunt funcționale	În general elementele proiectului sunt funcționale	Multe dintre elementele proiectului nu sunt funcționale
5. Grafică	Grafica jocului este adecvată, atractivă, focalizează atenția	Grafica jocului este adecvată	Grafica este atractivă, dar neadecvată scopului jocului	Grafica proiectului este inadecvată, neatractivă

B. Prezentarea proiectului

Criteria	10-9	8-7	6-5	4-3-2-1
1. Cunoașterea conținutului proiectului	Elevii cunosc complet conținutul proiectului.	Elevii cunosc în mare proporție conținutul proiectului	Elevii cunosc în oarecare măsură conținutul proiectului	Elevii cunosc foarte puțin din conținutul proiectului
2. Voce	Fiecare cuvânt rostit poate fi auzit clar și fără dificultăți de audiență Vorbitorul utilizează un volum/ton adecvat al vocii, are inflexiuni ale vocii, gesturi și expresii faciale care să atragă atenția auditoriului asupra proiectului. Prezentarea este convingătoare, denotă energie și entuziasm.	Fiecare cuvânt rostit poate fi auzit clar și fără dificultăți de audiență Vorbitorul face eforturi să fie clar, expresiv și să atragă atenția audienței.	Cea mai mare parte a prezentării poate fi auzită clar și fără dificultăți de audiență. Tonul și gesturile sunt inexpresive, plictisitoare.	Prezentatorul vorbește încet și neclar. Tonul este monoton. Lipsește energia și entuziasmul.
3. Ritm	Ritmul prezentării este natural, adecvat (nici prea lent pentru a fi plictisitor, nici prea alert pentru a nu putea fi înțeles conținutul proiectului). Nu există pauze inadecvate.	Ritmul prezentării este puțin prea lent sau puțin prea rapid.	Ritmul prezentării este prea lent sau prea rapid, apar pauze nejustificate în prezentare.	Prezentare nu are ritm adecvat, apar numeroase pauze în prezentare.
4. Contact vizual	Contactul vizual cu audiența este permanent	Nu este realizat permanent contactul vizual cu audiența	Contactul vizual cu audiența este slab	Nu este realizat un contact vizual cu audiența
5. Exersare	Este evident faptul că prezentarea a fost exersată, vorbitorii se sincronizează perfect. Nu sunt utilizate notițe sau materiale ajutătoare.	Prezentarea a fost exersată, vorbitorii cunosc structura prezentării. Nu sunt utilizate materiale ajutătoare.	Prezentarea nu a fost exersată suficient, vorbitorii nu se sincronizează, sunt utilizate notițe sau materiale ajutătoare.	Prezentarea nu a fost exersată
6. Lungimea prezentării	Prezentarea a fost realizată în timpul alocat	Prezentarea este puțin mai lungă/scurtă decât timpul alocat	Prezentarea este semnificativ mai lungă/scurtă față de timpul alocat	Prezentarea este mult prea lungă/scurtă față de timpul alocat

C. Evaluarea și autoevaluarea

Criteria	10-9	8-7	6-5	4-3-2-1
Evaluare/ autoevaluare	Pentru fiecare criteriu de evaluare notele sunt acordate în mod adecvat și sunt însoțite de justificări	Pentru majoritatea criteriilor de evaluare notele sunt acordate în mod adecvat și sunt însoțite de justificări	În general notele sunt acordate în mod adecvat	În cea mai mare parte notele nu sunt acordate adecvat și nu există justificări.

D. Consemnarea în blog-ul personal

Criteria	10-9	8-7	6-5	4-3-2-1
1. Conținut	Consemnarea din blog se referă la proiectul realizat și ilustrează modul în care realizarea acestui proiect a influențat dezvoltarea profesională și personală a elevului	Consemnarea din blog se referă la proiectul realizat și ilustrează modul în care realizarea acestui proiect a influențat dezvoltarea profesională, dar ilustrează mai puțin achizițiile în plan personal.	Consemnarea din blog se referă la proiectul realizat și ilustrează sumar în care realizarea acestui proiect a influențat dezvoltarea profesională/personală a elevului	Consemnarea din blog nu ilustrează modul în care realizarea acestui proiect a influențat dezvoltarea profesională și personală a elevului
2. Redactare conținut	Conținutul este corect redactat (sunt utilizate diacritice, nu conține erori gramaticale sau lexicale, sunt respectate regulile de tehnoredactare)	Conținutul este în mare măsură corect redactat	Conținutul este în general corect redactat	Conținutul conține numeroase greșeli de redactare
3. Exemple și detalii	Consemnarea conține numeroase exemple și detalii care ilustrează modul în care proiectul a contribuit la dezvoltarea profesională/personală a elevului.	Consemnarea conține suficiente exemple și detalii care ilustrează modul în care proiectul a contribuit la dezvoltarea profesională/personală a elevului.	Consemnarea conține puține exemple și detalii care ilustrează modul în care proiectul a contribuit la dezvoltarea profesională/personală.	Consemnarea nu conține exemple/detalii, este prea generală.
4. Formulare, creativitate	Textul este fluent, ușor de citit și de înțeles. Cuvintele sunt utilizate în mod adecvat, sunt expresive, uneori surprinzătoare, transmit cu claritate atât elementele obiective, cât și cele emoționale ale consemnării. Lectura consemnării este plăcută, atractivă. Textul conține elemente originale, creative care surprind cititorul în mod plăcut	Textul este fluent, ușor de citit/ înțeles. Cuvintele sunt utilizate în mod adecvat, transmit mesajul cu claritate. Alegerea cuvintelor nu este deosebit de expresivă, textul nu este foarte atractiv. Textul conține anumite elemente de creativitate care sporesc plăcerea cititorului	Textul este fluent, ușor de citit și de înțeles. Cuvintele sunt utilizate în mod adecvat, transmit cu claritate, dar sunt comune, nu au expresivitate și atractivitate. Foarte puține elemente de creativitate	Textul este dificil de înțeles, nu este fluent. Cuvintele utilizate sunt comune, inadecvate, nu exprimă cu claritate mesajul.

Notă pentru profesori:

La adresa <http://ler.is.edu.ro/~ema/jocuri.html> sunt publicate jocuri realizate de elevi în 2013 și 2020.

Considerații finale

Această activitate de învățare dezvoltă toate competențele din programa școlară a clasei a IX-a și poate fi în mod natural integrată la unitatea de învățare “Subprograme” din programa clasei a X-a. Activitatea poate fi reluată în spirală, măbind complexitatea jocului proiectat, fiind o activitate de învățare utilă pentru unitatea de învățare “Programarea orientată pe obiecte” din programa clasei a XI-a ce poate fi integrată cu succes în programa clasei a XII-a, în cadrul unității de învățare “Programare vizuală”.

Activitatea de învățare prezentată îmbină plăcerea jocului cu dorința naturală a elevilor de a crea propriile jocuri, prin urmare deosebit de atractivă și antrenantă pentru elevi și creează contextul perfect pentru dezvoltarea independenței în învățare, a responsabilității, a abilităților de organizare și de lucru în echipă, abilități esențiale pentru un profesionist în informatică. Realizarea unei aplicații software funcționale va fi un factor deosebit de motivant pentru elevi și va genera o atitudine pozitivă față de învățare pentru viitor.

Profesorul poate publica toate jocurile realizate de elevi, astfel încât activitatea se poate finaliza cu o sesiune de testare, în care fiecare elev să testeze jocurile create de colegi. Va fi o activitate amuzantă pentru elevi, dar în același timp e important ca elevii să ofere colegilor un feed-back constructiv. De multe ori o astfel de sesiune de testare se finalizează cu idei și soluții noi și reprezintă punctul de plecare pentru versiunea 2.0 a jocului. De asemenea, profesorul poate lansa o provocare și către părinți, să fie și ei testerii pentru jocul creat de propriul copil. Implicând părinții, aceștia vor conștientiza nivelul așteptărilor și progresul înregistrat de copilul lor în învățare, vor aprecia activitatea elevului și îl vor susține. Studiile demonstrează că evoluția elevilor are o pantă mai mare atunci când părinții sunt implicați în educația lor.

3. Activitatea de recuperare „Panglica”**Competențe specifice:**

IX.4.2/X.1.1 Identificarea necesității structurării datelor în tablouri

IX.4.3/X.3.1 Prelucrarea datelor structurate

IX.4.4 Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor

IX.4.5 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

X.1.3 Alegerea structurii de date adecvate rezolvării unei probleme

X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Enunț

Gigel are o panglică alcătuită din benzi de 1 cm lățime, colorate în diverse culori. Panglica are N benzi colorate cu C culori, culori pe care le vom numerota de la 1 la C . Gigel vrea ca la ambele capete ale panglicii să aibă aceeași culoare, dar cum nu poate schimba culorile benzilor, singura posibilitate rămâne tăierea unor bucăți de la capete.

Cerință

Scrieți un program care să determine modul de tăiere a panglicii astfel încât la cele două capete să fie benzi de aceeași culoare, iar lungimea panglicii obținute să fie maximă.

Date de intrare

Fișierul de intrare `panglica.in` conține pe prima linie numerele naturale N și C separate printr-un spațiu; pe următoarele N linii se află descrierea panglicii: pe fiecare linie un număr natural de la 1 la C , reprezentând în ordine culorile fâșiilor ce alcătuiesc panglica.

Date de ieșire

Fișierul de ieșire `panglica.out` va conține următoarele 4 numere: pe prima linie numărul de fâșii rămase; pe linia a doua numărul culorii care se află la capete; pe linia a treia câte fâșii trebuie tăiate de

la începutul panglicii inițiale; pe linia a patra câte fâșii trebuie tăiate de la sfârșitul panglicii inițiale.

Restricții

- $2 \leq N \leq 10000$
- $1 \leq C \leq 200$
- Dacă există mai multe soluții alegeți pe cea în care se taie cât mai puțin din partea de început a panglicii.

Exemple

panglica.in	panglica.out
6 3	4
1	2
2	1
1	1
3	
2	

panglica.in	panglica.out
5 2	4
1	2
2	1
1	0
2	
2	

Timp maxim de execuție pe test: 0.1 secunde

Memorie totală disponibilă: 2 MB, din care 1 MB pentru stivă.

Abordări posibile

În primul rând vom „dezbrăca” problema de poveste și o vom reformula, identificând modelul matematic din spatele problemei:

Se consideră un vector, în care sunt reținute N elemente cuprinse între 1 și C (culorile celor N benzi).

Să se determine o subsecvență de lungime maximă care să înceapă și să se termine cu aceeași valoare.

Dacă există mai multe soluții, se alege subsecvența cu poziția de început minimă.

Ideea 1

Construim toate subsecvențele posibile (există $N * (N+1) / 2$ astfel de subsecvențe), pentru fiecare subsecvență verificăm dacă pe poziția de început și pe poziția de sfârșit se află aceeași valoare și dacă da, comparăm lungimea subsecvenței cu lungimea maximă și o reținem dacă este mai mare.

Această abordare are complexitatea $O(N^2)$, deci vom depăși timpul maxim de execuție pentru o parte dintre testele de evaluare.

Desigur, ideea poate fi rafinată, putem analiza subsecvențele în ordinea descrescătoare a lungimilor lor (și crescător după poziția de început), astfel încât prima soluție găsită să fie optimă (dar și în acest caz avem o complexitate pătratică, se pot construi teste pentru care obținem depășire de timp).

Am putea prin urmare elimina această idee, fără a o implementa, deoarece N fiind mare, evident nu vom obține 100 de puncte.

Ideea 2

Observăm că N este mare, dar C (numărul de culori) este foarte mic. O altă observație ar fi că, pentru o anumită culoare x , subsecvența de lungime maximă având culoarea x la ambele capete se obține utilizând prima și ultima apariție a culorii x în vector. Utilizând aceste două observații, deducem că pentru fiecare culoare posibilă ne interesează să reținem poziția primei apariții și respectiv poziția ultimei apariții în vector a culorii respective. Pentru aceasta vom utiliza doi vectori `prim` și `ultim` având câte C elemente, pe care îi vom construi chiar de la citirea datelor.

Pentru a determina subsecvența de lungime maximă care începe și se termină cu aceeași culoare, parcurgem simultan cei doi vectori, determinăm lungimea subsecvenței pentru culoarea curentă i (`ultim[i]-prim[i]+1`) și o comparăm cu lungimea maximă lg_{max} . Dacă este mai mare, schimbăm lg_{max} , precum și c_{max} (culoarea pentru care se obține maximum). Dacă sunt egale, verificăm dacă poziția de început pentru culoarea i este mai mică decât poziția de început pentru c_{max} și dacă da, schimbăm c_{max} .


```
#include <fstream>
#define CMAX 201
using namespace std;
ifstream fin("panglica.in");
ofstream fout("panglica.out");

int prim[CMAX], ultim[CMAX];
int lgmax, cmax;
int main()
{
    int N, C, i, x;
    //citire date
    fin>>N>>C;
    for (i=1; i<=N; i++)
        {fin>>x;
        ultim[x]=i;
        if (prim[x]==0) prim[x]=i;
        }
    //determinare lungime maxima
    for (i=1; i<=C; i++)
        if (lgmax<ultim[i]-prim[i]+1)
            {lgmax=ultim[i]-prim[i]+1;
            cmax=i;
            }
        else
            if (lgmax==ultim[i]-prim[i]+1 && prim[i]<prim[cmax]) cmax=i;
    //afisare
    fout<<lgmax<<'\n'<<cmax<<'\n'<<prim[cmax]-1<<'\n'<<N-ultim[cmax]<<'\n';
    return 0;
}
```

Notă pentru profesori:

Dacă vom trimite pe arhiva educațională .campion această soluție, vom obține următorul borderou de evaluare:

The screenshot shows a web browser window with the URL `campion.edu.ro/arihiva/index.php?page=source&action=view&id=607414`. The page content includes:

- Concurent:** Emanuela Cerchez (emanuela_cerchez)
- Problemă:** panglica
- Limbaj:** cpp
- Data:** 2 iul 2020, 19:26
- Progres:** evaluare completă
- Punctaj exemplu:** 20
- Punctaj final:** 100
- Sursă:** vizualizare

Teste de evaluare:

Test	Timp	Mesaj evaluator	Punctaj
0	0	Okay!	10
1	0	Okay!	10
Total exemple			20
2	0	Okay!	10
3	0	Okay!	10
4	0	Okay!	10
5	0	Okay!	10
6	0	Okay!	10
7	0	Okay!	10
8	0	Okay!	10
9	0	Okay!	10
10	0	Okay!	10
11	0	Okay!	10
Total evaluare			100

Punctaje parțiale se pot obține pentru algoritmi corecți de complexități diferite, sau pentru diferite „bug”-uri (de exemplu, elevul uită să trateze cazul că pentru lungimi egale, se alege subsecvența cu poziția de început minimă).

În cazul în care sursa elevului are un punctaj < 100 puncte, elevul primește un „sprijin” în învățare (devine accesibilă legătura „Indicații de rezolvare”). Elevul poate studia indicațiile de rezolvare, își poate îmbunătăți soluția și o poate retrimite, ori de câte ori este necesar.

Profesorul poate consulta borderoul de evaluare pentru ultima sursă trimisă de elev și sursa elevului și poate oferi sprijin suplimentar, dacă este necesar.

Pe parcursul analizei problemei, elevii pot veni cu numeroase soluții algoritmice. Fiecare dintre acestea trebuie analizată, în primul rând din punctul de vedere al corectitudinii (dacă este corectă, justificăm corectitudinea, dacă este greșită, este suficient să oferim un contraexemplu, adică un set de date de intrare pe care algoritmul nu funcționează). Dacă soluția este corectă, aceasta va fi analizată și din punctul de vedere al eficienței (vom determina complexitatea timp/spațiu și verificăm dacă se încadrează în specificațiile problemei).

Este recomandat ca la sfârșitul activității să propunem spre rezolvare elevilor cel puțin o problemă similară. În funcție de nivelul elevilor, sarcinile de lucru pot fi individualizate. Iată, spre exemplu, exemple de sarcini de lucru similare, dar de niveluri diferite de dificultate:

- exemplul 1 (ușor): problema *SubNumar* (797) de pe Pbinfo (pbinfo.ro)
- exemplul 2 (mediu): problema *robot2* (3382) de pe Pbinfo (pbinfo.ro)
- exemplul 3 (dificil): problema *factori* de pe arhiva educațională .campion (campion.edu.ro/arihiva)

Clasa a XI-a (matematică-informatică, intensiv informatică)

Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar

În vederea elaborării planificării calendaristice se recomandă următorul demers:

- se analizează programele școlare în vigoare, clasele a X-a și a XI-a, disciplina Informatică
 - o programa clasei a X-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_in_tensiv_clasa%20a%20X_a.pdf
 - o programa clasei a XI-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_in_tensiv_clasa%20a%20XI-a.pdf
- din programa clasei a X-a se extrag acele competențe specifice a căror structurare ar fi trebuit realizată preponderent în semestrul al II-lea (în perioada în care învățarea s-a realizat la distanță) și conținuturile aferente;
- se analizează competențele specifice din programa clasei a XI-a și se stabilesc legături între acestea și competențele nestructurate/parțial structurate în anul școlar 2019-2020;
- se analizează conținuturile neabordate/parțial abordate din clasa a X-a și se stabilește modul de fuzionare cu elementele de conținut aferente clasei a XI-a, astfel încât logica internă a domeniului să fie păstrată în condițiile structurării competențelor specifice ale clasei a XI-a.

Prin această analiză se pot identifica unitățile de învățare care au fost desfășurate prin activități derulate exclusiv online. Competențele specifice corespunzătoare sunt X.1.1, X.1.2, X.1.3, X.2.3, X.3.1, X.3.3, X.3.4, X.4.1, X.4.2, X.5.1, X.5.2. În anexa 4 sunt prezentate aceste competențe și sunt reținute conținuturile, care sunt corelate cu parcursul școlar din perioada vizată de această analiză.

Competențele X.1.1 și X.1.3 asociate **Structurilor de date alocate dinamic** (Alocarea dinamică a memoriei – operații și mecanisme specifice) pot fi dezvoltate, în anul școlar 2020-2021, în timpul structurării competențelor XI.1.2 și XI.1.7. De exemplu, elemente din conținuturile legate de alocarea dinamică a memoriei vor fi reluate și aprofundate în cadrul noilor tipuri de date structurate introduse în clasa a XI-a – grafuri și structuri de date arborescente.

Competențele X.1.2 și X.3.1 „Prelucrarea datelor structurate”, vor fi dezvoltate în „spirală” în anul 2020-2021 în procesul de formare a competențelor XI.4, XI.5 și XI.6, pe măsură ce sunt introduse conținuturile legate de grafuri și arbori.

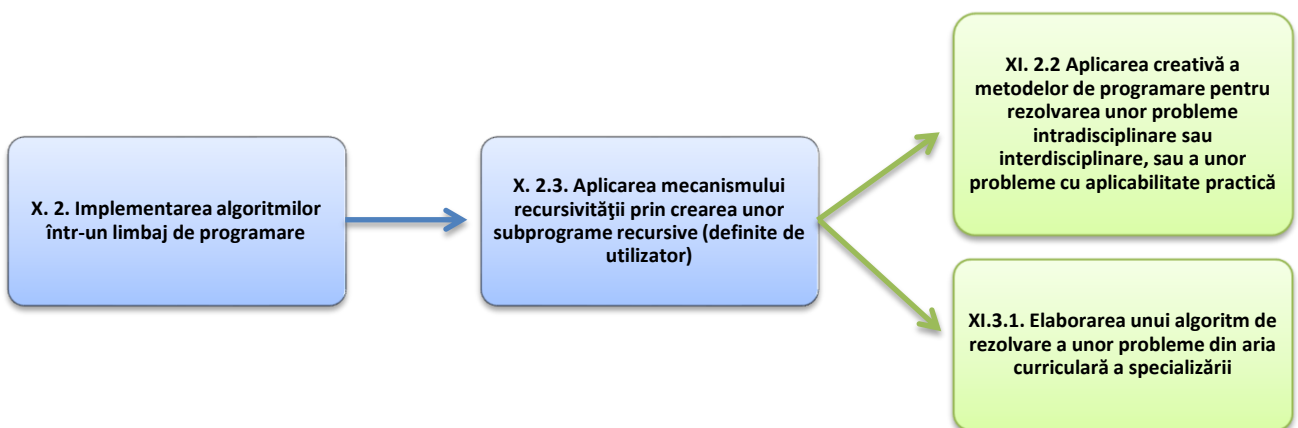
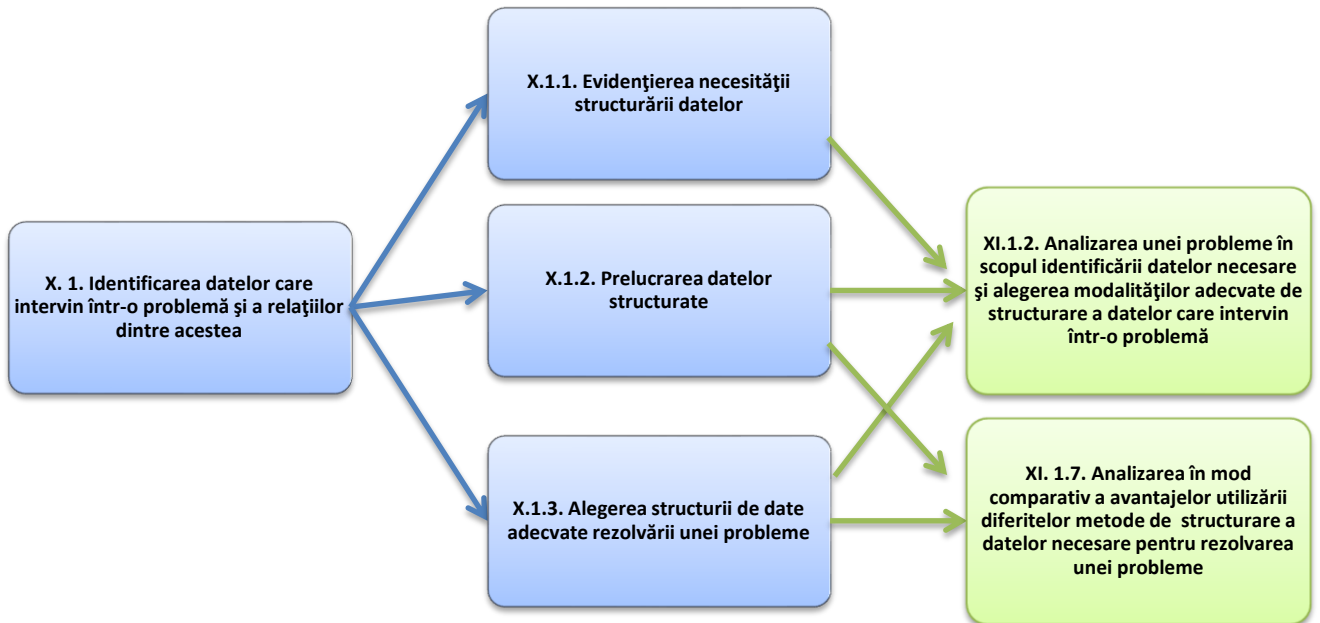
Competența X.2.3 „Aplicarea mecanismului recursivității prin crearea unor subprograme recursive definite de utilizator”, va fi antrenată și dezvoltată în anul 2020-2021 pentru obținerea competențelor XI.2.2 și XI.3.1 care vor face și conexiunea între problemele cu aplicabilitate practică și metodele/tehnicele de implementare a algoritmilor proiectați.

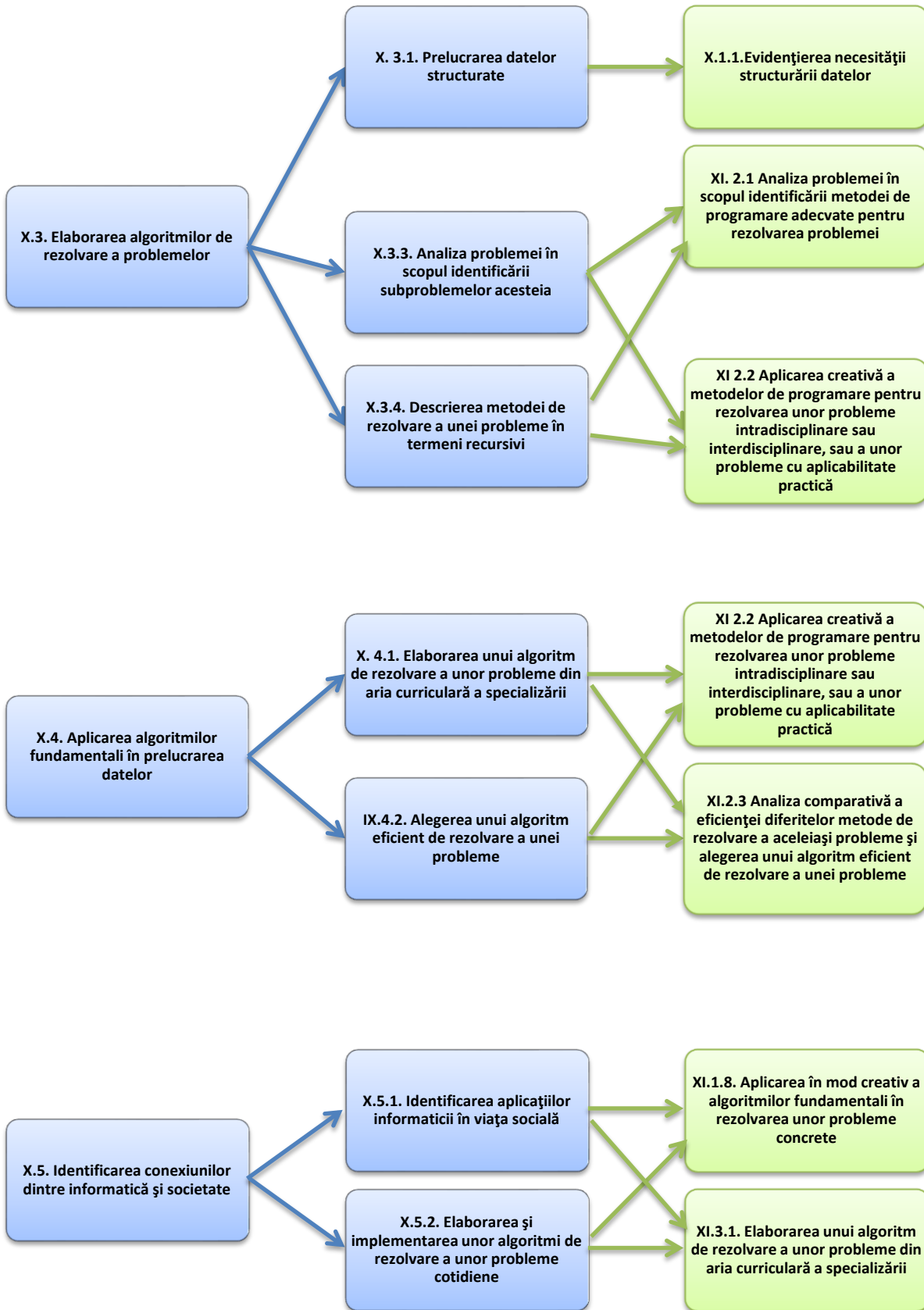
Competența X.3.3 “Analiza problemei în scopul identificării subproblemelor acesteia” va fi folosită succesiv în cadrul formării competențelor XI.2.1. și XI.2.2.

La competențele X.4.1 și X.4.2 se va apela, în anul școlar 2020-2021, în procesul de dezvoltare a competențelor XI.2.2 și XI.2.3 care dezvoltă la elevi capacitatea de proiectare, analiză și optimizare a algoritmilor de rezolvare a unor probleme din aria curriculară a specializării

Competențele X.5.1 și X.5.2, dezvoltate în spirală, de-a lungul a doi ani de studiu 2018-2020, vor fi antrenate în cadrul activităților pentru formarea competențelor XI.1.8 și XI.3.1. competențe care permit cunoașterea și aplicarea în mod creativ a algoritmilor fundamentali în rezolvarea unor probleme concrete, dar și analiza eficienței acestora.

Pentru atingerea tuturor competențelor, profesorul va încerca prioritizarea conținuturilor, dirijând învățarea în funcție de gradul de progres al elevilor. Pentru aceasta este necesară o evaluare inițială riguroasă, însoțită de activități remediale, acolo unde este necesar. În continuare este prezentată o modalitate în care se poate continua dezvoltarea competențelor specifice din clasa a X-a rămase nestructurate, de-a lungul clasei a XI-a, conform programei școlare de informatică filiera teoretică, profil real, specializarea Matematică-informatică intensiv informatică și filiera vocațională, profil militar, specializarea Matematică-informatică intensiv informatică.





Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior

În cadrul acestei secțiuni, exemplificăm itemi de evaluare prin care profesorul face o diagnoză a nivelului achizițiilor elevilor, prin raportare la următoarele competențe specifice: X.1.2, X.2.1, X. 2.2, X.2.3, X.3.1, X.3.3, X.3.4.

Item de evaluare inițială. Exemplul 1 – Nivel 1

Competența specifică

X.2.1 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator.

Enunț

Considerăm că tabloul unidimensional A memorează N elemente ($N < 100$), numere întregi. Subprogramul Del efectuează ștergerea tuturor elementelor egale cu x, din vectorul A. Identificați antetul corect al subprogramului Del.

- `int Del(int A[100], int N, x)`
- `void Del(int& A[100], int& N, int x)`
- `void Del(int A[100], int& N, int x)`
- `void Del(int A[100], int N, int x)`

Răspuns corect: c

Item de evaluare inițială. Exemplul 2 – Nivel 2

Competența specifică:

X.2.1 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator.

Enunț

Câte elemente divizibile cu 10 se vor afișa în urma executării programului alăturat?

- 4
- 3
- 2
- 1

Răspuns corect: b

```
#include <iostream>
using namespace std;
int F(int &y, int x){
    y = y/10 + x;
    return x + y;
}
int main()
{int x = 101, y = 10;
  cout<<F(x,y)<<" ";
  cout << x <<" "<< y<<" ";
  cout<<F(x,y);
  return 0 ; }
```

Item de evaluare inițială. Exemplul 3 – Nivel 2

Competența specifică:

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi.

Enunț

Se consideră subprogramul f, definit alăturat.

- Indicați ce valoare returnează apelul $f(17)$.
 - 3
 - 4
 - 6
 - 7

```
int f(int n)
{ if (n<10) return f(n+1)+3;
  if (n==10) return 7;
  return f(n-2)-1;
}
```

Răspuns corect: c

2. Dați exemplu de o valoare > 10 pe care o poate avea n , astfel încât $f(n)$ să returneze valoarea 7.

Răspuns corect: 15

Item de evaluare inițială. Exemplul 4 – Nivel 1-2

Competențe specifice

X.3.1 Prelucrarea datelor structurate (4a)

X.2.1 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator. (4b)

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia (4c)

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi (4d)

Notă pentru profesori:

Un item de evaluare vizează o singură competență specifică, dar acest item este compus de fapt din patru cerințe distincte, de aceea apar patru competențe distincte.

Enunț

Se consideră funcțiile `euclid()` și `calcul()` definite alăturat.

- Ce valoare va returna apelul `calcul(1,6)`, dacă tabloul unidimensional $A = \{0, 84, 98, 14, 42, 70, 56\}$?
- Dați un exemplu pentru tabloul A pentru care, în urma apelului `calcul(1,6)`, se va returna 1.
- Enunțați problema rezolvată de apelul `calcul(st,dr)`, menționând metoda de programare utilizată în implementarea funcției.
- Rescrieți subprogramul `euclid()` în manieră recursivă.

```
int euclid(int x,int y){
    int r;
    while (y)
    { r=x%y;
      x=y;
      y=r;
    }
    return x;
}

int calcul(int st,int dr)
{
    if (dr-st<=1)
        return euclid(A[st], A[dr]);
    int m = (st + dr)/2;
    return euclid(calcul(st, m),
                  calcul(m + 1, dr));
}
```

Răspuns corect 4a: 14

Răspuns corect 4b: Tabloul A va conține cel puțin 7 elemente, astfel încât cel mai mare divizor comun al elementelor $A[1], A[2], \dots, A[6]$ să fie egal cu 1.

Răspuns corect 4c:

Să se determine cel mai mare divizor comun al elementelor situate în tabloul unidimensional A pe pozițiile $st, st+1, \dots, dr$, utilizând metoda Divide et Impera.

Răspuns corect 4d:

O variantă de implementare recursivă ar putea fi :

```
int euclid(int x, int y)
{if (y)
    return euclid(y, x%y);
    return x;
}
```

Item de evaluare inițială. Exemplul 5 – Nivel 2

Competența specifică:

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi

Enunț:

Se consideră programul alăturat.

- Ce va afișa pe ecran programul pentru $n=5$?
- Ce reprezintă rezultatul returnat de apelul funcției $Ghici(p, q)$? Dar rezultatul afișat de program pentru un n dat?

```
#include <iostream>
using namespace std;

long long int Ghici(int p, int q)
{if (p>q) return 1;
 int r=(p+q)/2;
 return r*Ghici(p, r-1)*Ghici(r+1, q);
}

int main()
{int n;
 cout << "n= "; cin >> n;
 cout << Ghici(1,n);
 return 0;
}
```

Răspunsuri corecte:

- 120
- Apelul $Ghici(p, q)$ returnează produsul numerelor naturale din intervalul $[p, q]$. Programul afișează $n!$.

Item de evaluare inițială. Exemplul 6 – Nivel 2**Competența specifică:**

X.1.2 Prelucrarea datelor structurate

Enunț:

Variabila p memorează adresa de început a listei liniare nevide simplu înlănțuite L_1 . Elementele listelor sunt de același tip și memorează diverse informații utile, iar în câmpul adr rețin adresa elementului următor. Variabilele p și r sunt de același tip.

Dacă L_1 este organizată ca stivă, stabiliți care este operația corectă de extragere din stivă:

- | | |
|--------------------------------------|--------------------------------------|
| a. $r=p$; $p=p->adr$; delete r ; | c. $r=p->adr$; $p=r$; delete p ; |
| b. delete p ; $p=p->adr$; | d. $r=p->adr$; $p=r$; delete r ; |

(Bacalaureat iulie 2003)

Răspuns corect: a**Item de evaluare inițială. Exemplul 7 – Nivel 1****Competența specifică:**

X.1.2 Prelucrarea datelor structurate

Enunț:

Care dintre următoarele variante constituie declararea în limbajul C/C^{++} a unui tip de date care reprezintă un nod dintr-o listă simplu înlănțuită care conține un caracter, precum și adresa următorului nod din listă.

- struct Nod {char inf; Nod urm;};
- struct Nod {char inf; Nod* urm;};
- struct Nod {char inf; Nod& urm;};
- struct Nod {char inf; char* urm;};

Răspuns corect: b

Item de evaluare inițială. Exemplul 8 – Nivel 1**Competența specifică**

X.1.2 Prelucrarea datelor structurate.

Enunț

Considerăm declarația următorului tip de date necesar reprezentării unei liste simplu înlănțuite:

```
struct nod
```

```
{ int inf; nod *leg; };
```

Funcția V_m , declarată alăturat, primește, prin intermediul parametrului p , adresa de început a unei liste simplu înlănțuite.

```
int Vm(nod *p)
{ if (p==NULL) return 1;
  if (p->inf>0)
    return Vm(p->leg);
  return 0;
}
```

Funcția V_m va returna valoarea 1 dacă și numai dacă:

- lista este vidă;
- lista are primul element pozitiv;
- toate valorile memorate în listă sunt strict pozitive;
- toate valorile memorate în listă sunt nenule.

Răspuns corect: c**Item de evaluare inițială. Exemplul 9 – Nivel 3**

Propunem ca exemplu problema *Imagine* de pe arhiva educațională .campion:

(<http://campion.edu.ro/arhiva/index.php?page=problem&action=view&id=771>).

Competența specifică:

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia

Enunț

Să considerăm o imagine alb-negru de dimensiune $L \times L$ pixeli. Un pixel poate fi alb (codificat cu valoarea 0) sau negru (codificat cu valoarea 1). Imaginile pot fi compresate în diverse moduri. Una dintre cele mai cunoscute scheme de compresie este următoarea:

1. Dacă imaginea este formată atât din pixeli 1, cât și din pixeli 0, se reține valoarea 1, care indică faptul că imaginea va fi partiționată în alte 4 subimagini, așa cum este descris la pasul 2. Altfel codificăm întreaga imagine ca 00 sau 01 semnificând faptul că întreaga imagine este formată numai din pixeli 0, respectiv numai din pixeli 1.

2. O imagine **I** este împărțită în 4 subimagini **A**, **B**, **C**, **D** după cum este ilustrat în figura următoare:



Apoi se aplică din nou pasul 1, pentru fiecare dintre cele 4 subimagini, în ordinea **A**, **B**, **C**, **D**. Numărul de biți (cifre de 0 sau 1) obținuți în urma compresiei reprezintă dimensiunea imaginii compresate.

Cerință

Dată fiind o imagine, să se determine dimensiunea imaginii compresate.

Date de intrare

Fișierul de intrare `image.in` conține pe prima linie L (dimensiunea imaginii). Următoarele L linii conțin imaginea codificată, fiecare linie conținând exact L valori 0 sau 1 separate prin câte un spațiu.

Date de ieșire

Fișierul de ieșire `image.out` conține o singură linie pe care se află un număr natural care reprezintă dimensiunea imaginii compresate.

Restricții

$1 \leq L \leq 250$

Dacă zona **I** are colțul stânga-sus (x_s, y_s) și colțul dreapta-jos (x_d, y_d) , atunci colțul din dreapta-jos al zonei **B** va fi $((x_s+x_d)/2, (y_s+y_d)/2)$.

Exemple

image.in	image.out	Explicații
<pre>4 1 1 1 1 1 1 0 1 0 1 0 0 0 0 1 1</pre>	30	<p>Compresia imaginii din exemplul 1 se realizează astfel:</p> $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} = 1 \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} =$ $= 1 \quad 1(0) \quad (0) \quad (0) \quad (1) \quad 01 \quad 1(1) \quad (0) \quad (1) \quad (0) \quad 1(0) \quad (1) \quad (1) \quad (1) \quad =$ $= 1 \quad 1 \quad 00 \quad 00 \quad 00 \quad 01 \quad 01 \quad 1 \quad 01 \quad 00 \quad 01 \quad 00 \quad 1 \quad 00 \quad 01 \quad 01 \quad 01$ <p>Deoarece după compresare sunt necesari 30 de biți, dimensiunea imaginii compresate este 30.</p>
image.in	image.out	Explicații
<pre>8 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1</pre>	9	<p>Compresia imaginii din exemplul 2 se realizează astfel:</p> $\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = 1 \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} =$ $= 1 \quad 01 \quad 00 \quad 01 \quad 01$ <p>În acest caz dimensiunea imaginii compresate este 9.</p>

Timp maxim de execuție pe test: 0.1 secunde

Memorie totală disponibilă: 2 MB, din care 1 MB pentru stivă.

Răspuns corect: Vezi activitatea remedială "Imagine".

Item de evaluare inițială. Exemplul 10 – Nivel 3

Propunem ca exemplu problema `FStergeDuplicate` de pe platforma `pbinfo`

(<https://www.pbinfo.ro/probleme/3527/fstergeduplicate>).

Competența specifică:

X.1.2 Prelucrarea datelor structurate

Enunț

Se consideră o listă liniară simplu înlănțuită, alocată dinamic, în care elementele sunt de tipul declarat mai jos:

```
struct Nod
{ int val;
  Nod *next;
};
```

în care câmpul `val` memorează un număr întreg, iar câmpul `next` memorează adresa următorului element al listei. Valorile din listă sunt ordonate crescător.

Cerință

Să se scrie o funcție C++ cu următorul prototip:

```
void FStergeDuplicate(Nod *&L);
```

Funcția primește ca parametru un pointer `L` la o listă simplu înlănțuită care are valorile din noduri în ordine crescătoare. Funcția trebuie să elimine nodurile care conțin valori duplicate. De exemplu, dacă lista conține valorile `3, 3, 3, 5, 6, 7, 7, 7, 7, 20`, atunci după eliminarea duplicatelor, lista va conține `3, 5, 6, 7, 20`.

Restricții și precizări

Numele funcției va fi `FStergeDuplicate`.

Important

Soluția propusă va conține definiția funcției cerute. Prezența în soluție a altor instrucțiuni poate duce erori de compilare sau de execuție care vor avea ca efect depunctarea soluției.

Răspuns corect: Vezi activitatea remedială "`FStergeDuplicate`".

Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare

3.1. EXEMPLE DE ACTIVITĂȚI REMEDIALE

Activitatea remedială „Apeluri”

Item de evaluare inițială. Exemplul 1 – Nivel 1

Competența specifică:

X.2.1 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator

Notă pentru profesori:

Activitatea va viza înțelegerea modului de execuție a subprogramelor și a modului de efectuare a transferurilor de date. Activitatea se va concretiza sub forma unei conversații euristice, având la bază o succesiune de întrebări, plecând de la un subprogram realizat în limbajul C⁺⁺.

Activitatea poate fi desfășurată online, folosind instrumente de comunicare sincronă, dar și platforme care permit realizarea testelor (Google Forms, Kahoot, Quizizz, etc).

FIȘĂ DE LUCRU

Considerăm următoarea secvență de program:

```
int n=32;
int Det(int &x) {
    int s=0;
    while (x!=0) {
        s+=x%2;
        x/=2;
    }
    return s;
}
```

Întrebare	Răspuns așteptat
Care este antetul funcției Det () ?	int Det(int &x)
Care sunt variabile locale ale funcției Det () ?	Variabila întregă s
Care sunt parametrii funcției Det () ? Menționați modul de transmitere a parametrilor.	Parametrul x, transmis prin referință
Care este rezultatul returnat de apelul Det(n), pentru n=32?	1
Ce se va afișa în urma executării instrucțiunii următoare, pentru n=32? cout<<Det(n)+Det(n)+1;	2
Dacă am modifica antetul funcției Det transmițând parametrul x prin valoare, ce valoare va fi afișată în urma executării instrucțiunii următoare pentru n=32? cout<<Det(n)+Det(n)+1;	3

Activitatea remedială „Număr divizori”

Item de evaluare inițială. Exemplul 2 – Nivel 2

Competența specifică:

X.2.1 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a urmări modul de execuție “linie cu linie” a unui program, respectiv utilizarea corectă a subprogramelor definite de utilizator.

Activitatea propusă constă în rezolvarea unei fișe de lucru ce conține itemi de completare. Activitatea va fi individuală și poate fi desfășurată în clasă, precum și online.

FIȘĂ DE LUCRU

Funcția `Nd()` trebuie să determine numărul de divizori naturali ai unui număr natural nenul transmis prin parametrul `x`. Se consideră următoarea definiție incompletă a funcției `Nd()`:

```
... Nd(...) {
    int d, nr=0;
    for (d=1; d*d<x; d++)
        if (x%d==0) nr+=2;
    if (d*d==x) nr++;
    return ...;
}
```

Întrebare	Răspuns așteptat
Conform enunțului, funcția primește ca parametru un număr natural x . Cum trebuie declarat parametrul corespunzător?	<code>int x</code> sau <code>unsigned int x</code>
Funcția trebuie să calculeze numărul de divizori naturali ai parametrului <code>x</code> . Ce tip trebuie să aibă rezultatul returnat de funcție?	<code>int</code> sau <code>unsigned int</code>
În ce variabilă a fost calculat numărul de divizori în corpul funcției <code>Nd()</code> de mai sus?	<code>nr</code>
Care este definiția completă a funcției <code>Nd()</code> ?	<pre>int Nd(int x) { int d, nr=0; for (d=1; d*d<=x; d++) if (x%d==0) nr+=2; if (d*d==x) nr++; return nr; }</pre>
Completați punctele de suspensie astfel încât instrucțiunea următoare să afișeze valoarea 3. <code>cout<< Nd(...);</code>	Se poate completa cu orice număr natural care este pătratul unui număr prim. De exemplu, 25 are ca divizori pe 1, 5 și 25.

Activitatea remedială „Pare recursiv”**Item de evaluare inițială. Exemplul 3 – Nivel 2****Competența specifică:**

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi.

Notă pentru profesori:

Activitatea va urmări dezvoltarea competenței de rezolvare a unei probleme în termeni recursivi, precum și utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator. Întreaga activitate poate fi desfășurată și online, folosind instrumente de comunicare sincrone de comunicare cu elevii (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc), profesorul având posibilitatea prezentării itemului și rezolvării acestuia.

FIȘĂ DE LUCRU

1. Pentru a vă reaminti modul în care funcționează mecanismul recursivității, urmăriți tutorialul următor: <http://advancedelearning.com/index.php/articles/214>.
2. Fie următorul subprogram recursiv:



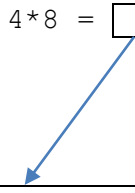
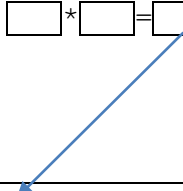
```
int f(int i)
{if (i==0) return 1;
  if (a[i]%2==0)
    return a[i]*f(i-1);
  return 0;
}
```

Urmăriți pas cu pas mecanismul recursivității pentru apelul $f(n)$ dacă $n=4$ și $a=\{0, 4, 8, 12, 6\}$ și completați în al doilea dintre tabelele următoare valorile corespunzătoare.

Apelurile recursive

Pasul	Parametrul i	Observații	Cod
1	4	$i=4$ $i==0?$ NU $a[i]\%2==0?$ DA ($6\%2==0?$) Apel $f(3)$	<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); return 0; }</pre>
2	3	$i=3$ $i==0?$ NU $a[i]\%2==0?$ DA ($12\%2==0?$) Apel $f(2)$	<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); return 0; }</pre>
3	2	$i=2$ $i==0?$ NU $a[i]\%2==0?$ DA ($8\%2==0?$) Apel $f(1)$	<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); return 0; }</pre>
4	1	$i=1$ $i==0?$ NU $a[i]\%2==0?$ DA ($4\%2==0?$) Apel $f(0)$	<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); return 0; }</pre>
5	0	$i=1$ $i==0?$ DA	<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); else return 0; }</pre>

Revenirea din apeluri

Parametrul i	Valoare returnată	Observații	Cod
0	1 		<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); else return 0; }</pre>
1	$1*4 = 4$ 		<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); else return 0; }</pre>
2	$4*8 = \square$ 		<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); else return 0; }</pre>
3	$\square * \square = \square$ 		<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); else return 0; }</pre>
4	$\square * \square = \square$		<pre>int f(int i) { if (i==0) return 1; if (a[i]%2==0) return a[i]*f(i-1); else return 0; }</pre>

Valoarea returnată este deci

3. Răspundeți la următoarele întrebări:

Întrebare	Răspuns așteptat
a. Ce valoare va returna apelul $f(0)$?	1
b. Ce valoare va returna apelul $f(4)$ dacă vectorul a este $a=\{0, 2, 10, 4, 6\}$?	$1*2*10*4*6=420$
c. Ce valoare va fi returnată în urma apelului $f(4)$ dacă vectorul a este $a=\{0, 2, 15, 4, 6\}$?	0
d. Deduceți folosindu-vă de exemplele de la punctul b) și c) ce valori trebuie să conțină vectorul a pentru ca funcția să returneze o valoare nenulă?	Vectorul trebuie să conțină doar valori pare nenule.

Activitatea remedială „Euclid”

Item de evaluare inițială. Exemplul 4 – Nivel 2

Competențe specifice:

X.2.1 Utilizarea corectă a subprogramelelor predefinite și a celor definite de utilizator.

X.2.3 Aplicarea mecanismului recursivității prin crearea unor subprograme recursive (definite de utilizator)

X.3.1 Prelucrarea datelor structurate

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a urmări execuția “linie cu linie” a unui program, respectiv utilizarea corectă a subprogramelor definite de utilizator în manieră iterativă sau recursivă.

Activitatea propusă constă în rezolvarea unei fișe de lucru care conduce elevul pas cu pas în rezolvarea sarcinii de lucru. Activitatea va fi individuală și poate fi desfășurată în clasă, dar și online.

FIȘĂ DE LUCRU

Considerăm funcția euclid() care determină cel mai mare divizor comun al numerelor naturale transmise prin parametrii x și y, în implementare iterativă.

```
int euclid(int x,int y){
int r;
while (y)
    { r=x%y;
    x=y;
    y=r;
    }
return x;
}
```

1. Urmăriți execuția pas cu pas a apelului euclid(106,16). Completați ultimele linii ale tabelul următor.

Pasul	Operația
1	x y cât rest 106 = 16 × 6 + 10
2	x y cât rest 16 = 10 × 1 + 6
3	x y cât rest 10 = 6 × 1 + 4
4	x y cât rest □ = □ × □ + □
5	x y cât rest □ = □ × □ + 0
6	x y cât rest □ = □ 0

Cel mai mare divizor comun determinat în acest fel este □ adică de ultimul rest diferit de 0.

2. Descrieți o implementare recursivă a funcției euclid(), valorificând câteva dintre concluziile obținute în urma execuției pas cu pas a apelului euclid(106,16).

Întrebare	Răspuns așteptat
Observăm că cel mai mare divizor comun al numerelor 106 și 16 coincide cu cel mai mare divizor comun al numerelor 16 și 10. Adică $cmmdc(x, y) = cmmdc(y, x \% y)$ dacă $y \neq 0$ Ce alte perechi de numere au același cmmdc cu 106 și 16, conform execuției acestui apel?	10 6 6 4 4 2

Care este cel mai mare divizor comun al numerelor a și 0 (a≠0)?	a
<p>Completați definiția recursivă a funcției euclid():</p> <pre>int euclid(int x, int y) {if () return euclid(); return ; }</pre>	<pre>int euclid (int x, int y) {if (y) return euclid (y,x%y); return x; }</pre>

3. Se consideră următoarea funcție `calcul()`, în care funcția `euclid()` este descrisă anterior.

```
int calcul(int st,int dr)
{ int m;
  if (dr-st<=1)
    return euclid(A[st], A[dr]);
  m = (st + dr)/2;
  return euclid(calcul(st, m), calcul(m + 1, dr));
}
```

Completați în tabelul de mai jos pașii efectuați de algoritm pentru apelul `calcul(1,6)` dacă tabloul unidimensional $A=\{0, 84, 98, 14, 42, 70, 56\}$?

<code>calcul(1,6)</code> = <code>euclid(calcul(1,3), calcul(4,6))</code> = <code>euclid(,)</code> =
<code>calcul(1,3)</code> = <code>euclid(calcul(1,2), calcul(3,3))</code> = <code>euclid(,)</code> =
<code>calcul(1,2)</code> = <code>euclid(A[1],A[2])</code> = <code>euclid(84,98)</code> =
<code>calcul(3,3)</code> = <code>euclid(,)</code> =
<code>calcul(4,6)</code> = <code>euclid(,)</code> = <code>euclid(,)</code> =
<code>calcul(4,5)</code> = = 14
<code>calcul(6,6)</code> = = =

În concluzie `calcul(1,6)` =

4. Ce reprezintă valoarea calculată la pasul anterior?

Răspuns corect:

Cel mai mare divizor comun al numerelor memorate în vectorul A pe pozițiile 1, 2, ..., 6.

5. Dați un exemplu de vector pentru care apelul `calcul(1,4)` va returna valoarea 1!

Activitatea remedială „Produs prin Divide et Impera”

Item de evaluare inițială. Exemplul 5 – Nivel 2

Competența specifică:

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi

Notă pentru profesori:

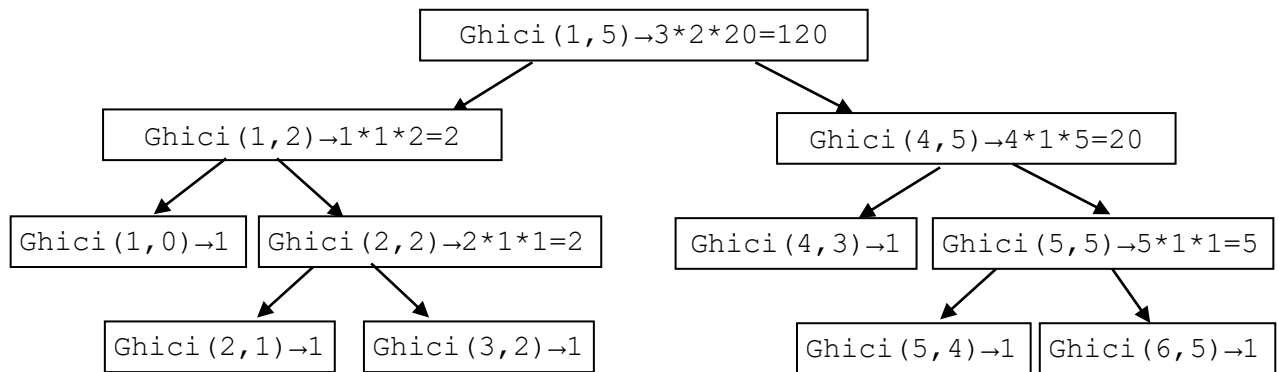
Propunem o activitate remedială, în trei etape:

- *Etapa 1*: urmărirea execuției pas cu pas a programului pentru $n=5$;
- *Etapa 2*: analiza programului în scopul identificării problemei rezolvate de către acesta;
- *Etapa 3*: propunerea spre rezolvare a unui alt exercițiu, de același tip.

FIȘĂ DE LUCRU

Etapa 1

Urmăriți execuția pas cu pas a apelului $Ghici(1, 5)$ ilustrat de arborele de execuție de mai jos:



Pentru o mai bună înțelegere a modului de funcționare a recursivității, urmăriți, în paralel, și modul în care se alocă memorie pe stivă pentru aceste apeluri.

Ilustrați arborele de execuție pentru apelul $Ghici(5, 8)$. Ce valoare returnează acest apel?

Răspuns corect: 1680

Etapa 2

Întrebare	Răspuns așteptat
Ce reprezintă valoarea returnată de apelul $Ghici(1, 5)$?	$120 = 1*2*3*4*5 = 5!$
Ce reprezintă valoarea returnată de apelul $Ghici(1, n)$?	$1*2*3*...*n = n!$
Ce reprezintă valoare returnată de apelul $Ghici(5, 8)$?	$5*6*7*8$
Ce reprezintă valoarea returnată de apelul $Ghici(p, q)$?	$Ghici(p, q)$ returnează produsul tuturor numerelor naturale din intervalul $[p, q]$.

Etapa 3. Exercițiu remedial de același tip

Se consideră următorul program:

```

#include <fstream>
#include <iostream>
#define NMax 100
using namespace std;
int a[NMax], n;
void Citire()
{ifstream fin("date.in"); int i;
  fin>>n;
  for (i=0; i<n; i++) fin>>a[i];
  fin.close();
}

int rez(int x, int y)
{int r;
  while (y) {r=x%y; x=y; y=r;}
  return x;
}
  
```

```

int Ghici(int p, int q)
{
    if (q-p <= 1) return rez(a[p], a[q]);
    int m=(p+q)/2;
    return rez(Ghici(p, m), Ghici(m+1, q));
}
int main ()
{Citire();
  cout<<Ghici(0, n-1)<<'\n';
  return 0;
}

```

a. Ce va afișa pe ecran programul dacă fișierul de intrare `date.in` conține:

```

5
495 720 4500 900 585

```

b. Ce reprezintă rezultatul afișat de program?

Activitatea remedială „Eliminarea unui element dintr-o stivă”

Item de evaluare inițială. Exemplul 6 – Nivel 2

Competența specifică:

X.1.2 Prelucrarea datelor structurate

Notă pentru profesori:

Propunem o activitate remedială, în trei etape:

- *Etapa 1:* reactualizarea cunoștințelor;
- *Etapa 2:* analiza fiecărei variante de răspuns, în scopul identificării erorilor/variantei corecte;
- *Etapa 3:* propunerea spre rezolvare a unor exerciții similare pentru a verifica dacă învățarea s-a produs.

FIȘĂ DE LUCRU

Etapa 1.

Stiva este o structură de date abstractă care suportă două operații: inserarea unui nod la începutul listei (PUSH) și ștergerea primului nod din listă (POP).

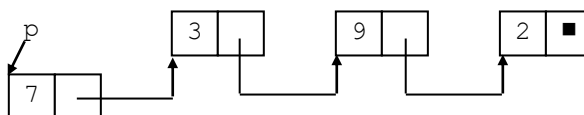
Stiva poate fi implementată în diferite moduri, inclusiv ca listă simplu înlănțuită.

În cazul nostru particular, trebuie să extragem nodul indicat de variabila `p`.

Utilizați animația de la adresa <http://www.cs.armstrong.edu/liang/animation/web/Stack.html> pentru a exersa operațiile PUSH și POP, plecând de la o stivă inițial vidă.

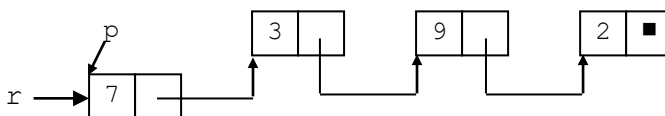
Etapa 2. Analiza variantelor

Urmăriți pas cu pas operațiile corespunzătoare fiecărei variante, desenând lista după execuția fiecărei instrucțiuni, pe următorul exemplu:

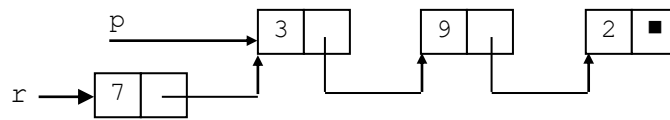


Varianta a.

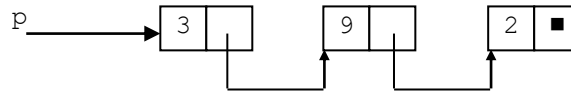
`r=p;`



`p=p->adr;`



delete r;



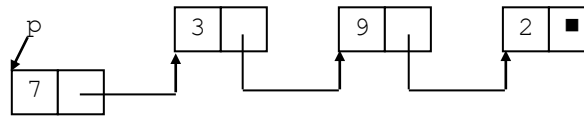
Așadar varianta a. este cea corectă.

Varianta b.

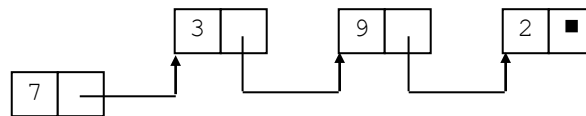
delete p; p=p->adr;

Prima instrucțiune din varianta b. eliberează zona de memorie alocată primului nod al listei, ca urmare nu mai putem accesa $p \rightarrow \text{adr}$, deci nici celelalte elemente ale listei.

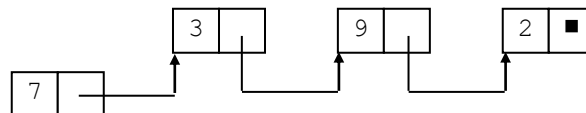
Varianta c.



$r = p \rightarrow \text{adr};$



$p = r;$



Varianta d.

$r = p \rightarrow \text{adr}; p = r; \text{delete } r;$

Comparați această variantă cu varianta c). Ce observați?

Primele două instrucțiuni coincid cu cele de la varianta c. Pointerii p și r indică același nod, respectiv al doilea nod din lista deci și varianta d. este greșită.

Etapa 3. Exerciții

Variabilele p și q memorează adresele de început ale listelor liniare nevide simplu înlănțuite L_1 și respectiv L_2 . Elementele listelor sunt de același tip și memorează diverse informații utile, iar în câmpul adr rețin adresa elementului următor. Variabilele p , q , r și u sunt de același tip.

1. Dacă L_1 este o coadă, cu p adresa primului și u adresa ultimului element, iar r adresa unui element ce urmează a fi adăugat în coadă, stabiliți care dintre următoarele este o operație corectă de adăugare

- | | |
|---|---|
| a. $r \rightarrow \text{adr} = u; u = r;$ | c. $r \rightarrow \text{adr} = p; p = r;$ |
| b. $r = u \rightarrow \text{adr}; u = r;$ | d. $u \rightarrow \text{adr} = r; u = r;$ |

2. Trebuie să fie mutat primul element al listei L_1 imediat după primul element al listei L_2 , în rest listele rămânând neschimbate. Care dintre atribuirile următoare sunt necesare și în ce ordine se efectuează?

1) $r=q \rightarrow \text{adr}$ 2) $r=p \rightarrow \text{adr}$ 3) $q \rightarrow \text{adr}=p$ 4) $p \rightarrow \text{adr}=r$ 5) $p=r$ 6) $p \rightarrow \text{adr}=q \rightarrow \text{adr}$

- a. 2 3 6 c. 1 6 3 5
 b. 1 3 4 d. 2 6 3 5

3. Pentru a uni listele L_1 și L_2 plasând lista L_1 în continuarea listei L_2 , se efectuează operațiile:

- a. $r=q; \text{while}(r \rightarrow \text{adr}) r=r \rightarrow \text{adr}; r \rightarrow \text{adr}=p;$
 b. $r=p; \text{while}(r \rightarrow \text{adr}) r=r \rightarrow \text{adr}; r \rightarrow \text{adr}=q;$
 c. $r=p; \text{while}(r) r=r \rightarrow \text{adr}; r=q;$
 d. $r=q; \text{while}(r \rightarrow \text{adr}) \{ r=r \rightarrow \text{adr}; r \rightarrow \text{adr}:=p; \}$

(Bacalaureat iulie 2003)

Activitatea remedială „Declarare tip de date – nod listă simplu înlănțuită”

Item de evaluare inițială. Exemplul 7 – Nivel 1

Competența specifică:

X.1.2 Prelucrarea datelor structurate

Notă pentru profesori:

Propunem o activitate remedială, în trei etape:

- Etapa 1: reactualizarea cunoștințelor;
- Etapa 2: analiza fiecărei variante de răspuns, în scopul identificării erorilor/variantei corecte;
- Etapa 3: propunerea spre rezolvare a unor exerciții similare.

FIȘĂ DE LUCRU

Etapa 1. Reactualizarea cunoștințelor

O **listă simplu înlănțuită** este o structură de date constituită dintr-o succesiune de elemente denumite noduri. Fiecare nod din listă conține două părți: o parte de informație (în care sunt memorate informațiile corespunzătoare nodului, specifice problemei) și o parte de legătură (în care este memorată adresa următorului element din listă).

Întrebare	Răspuns așteptat
Ce tip de date este adecvat atunci când dorim să memorăm în aceeași structură date de tipuri diferite?	struct
Ce tip de date utilizăm pentru a reține adresa unei zone de memorie?	pointer
Cum se declară un pointer p care reține adresa unei zone de memorie în care se află o dată de tipul Tip ?	<code>Tip * p;</code>
Cum se declară un tip de date care reprezintă un nod din listă va fi (unde Tip reprezintă tipul informației memorate în nodul respectiv):	<code>struct Nod { Tip inf; struct Nod * urm;};</code>

Etapa 2. Analiza variantelor de răspuns

Observăm că toate variantele de răspuns declară un struct denumit `Nod`, câmpul de informație fiind de tip `char`. Diferențele între variante apar la declararea câmpului care reține adresa următorului nod din listă. Ce reprezintă câmpul `urm` declarat în fiecare dintre variante?

Varianta	Răspuns așteptat
a.	O dată de tip <code>Nod</code> . Declarația este incorrectă sintactic.
b.	Adresa unei zone de memorie în care se află o dată de tip <code>Nod</code> (pointer). Varianta corectă.
c.	O referință la o dată de tip <code>Nod</code> .
d.	Adresa unei zone de memorie în care se află o dată de tip <code>char</code> (pointer).

Etapa 3. Exerciții

1. Declarați un tip de date care reprezintă un nod dintr-o listă simplu înlănțuită care conține ca informație un număr întreg, precum și adresa următorului nod din listă.
2. Declarați un tip de date care reprezintă un nod dintr-o listă simplu înlănțuită care conține ca informație un șir de maximum 20 de caractere, precum și adresa următorului nod din listă.

Activitatea remedială „Listă recursivă”

Item de evaluare inițială. Exemplul 8 – Nivel 1

Competența specifică:

X.1.2 Prelucrarea datelor structurate

FIȘĂ DE LUCRU

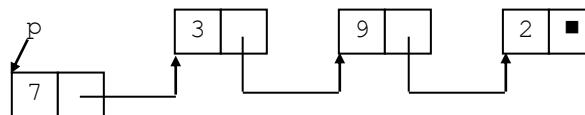
1. Considerăm declarația următorului tip de date necesar reprezentării unei liste simplu înlănțuite:

```
struct nod
{ int inf; nod *leg; };
```

Funcția `Vm`, declarată alăturat, primește, prin intermediul parametrului `p`, adresa de început a unei liste simplu înlănțuite.

```
int Vm(nod *p)
{ if (p==NULL) return 1;
  if (p->inf>0)
    return Vm(p->leg);
  return 0;
}
```

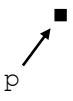
Urmăriți pas cu pas execuția apelului `Vm(p)`, considerând că `p` memorează adresa primului nod al listei din figura următoare.



Completați pașii 3 și 4 de mai jos după modelul pașilor anteriori.

Apelurile recursive

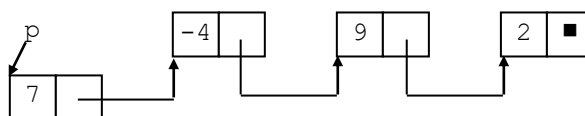
Pasul	Lista	Observații	Cod
1		<p><code>p==NULL?</code> NU <code>p->inf>0</code> DA (7>0?)</p> <p>Apel <code>Vm(p->leg)</code></p>	<pre>int Vm(nod *p) { if (p==NULL) return 1; if (p->inf>0) return Vm(p->leg); return 0; }</pre>
2		<p><code>p==NULL?</code> NU <code>p->inf>0</code> DA (3>0?)</p> <p>Apel <code>Vm(p->leg)</code></p>	<pre>int Vm(nod *p) { if (p==NULL) return 1; if (p->inf>0) return Vm(p->leg); return 0; }</pre>
3			<pre>int Vm(nod *p) { if (p==NULL) return 1; if (p->inf>0) return Vm(p->leg); return 0; }</pre>
4			<pre>int Vm(nod *p) { if (p==NULL) return 1; if (p->inf>0) return Vm(p->leg); return 0; }</pre>

5		$p == \text{NULL}?$ DA	<pre>int Vm(nod *p) { if (p==NULL) return 1; if (p->inf>0) return Vm(p->leg); return 0; }</pre>
---	---	------------------------	--

Revenirea din apeluri

Dacă urmăriți acum cu atenție tabelul anterior în sens invers vedeți că valoarea 1 se returnează recursiv până la sfârșit. Valoarea returnată este deci

2. Urmăriți pas cu pas, în același mod, execuția apelului $V_m(p)$ considerând că p memorează adresa primului nod al listei din figura următoare.



Ce valoare se returnează în urma apelului $V_m(p)$ în acest caz?

Răspuns corect: 0

În ce situație în urma apelului $V_m(p)$ se va returna valoarea 0?

Răspuns corect: O listă care conține cel puțin un element negativ sau egal cu 0.

Activitatea remedială „Imagine”

Item de evaluare inițială. Exemplul 9 – Nivel 3

Competența specifică:

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia

FIȘĂ DE LUCRU

Etapa 1. Analiză

Imaginea este o matrice pătratică cu latura L . Pe parcursul împărțirii matricei în zone conform enunțului, deoarece L nu este obligatoriu o putere a lui 2, se vor obține zone care nu sunt obligatoriu de formă pătrată. Ca urmare putem formula o subproblemă a problemei date astfel: "să se determine dimensiunea compresiei unei zone a matricei date".

Întrebare	Răspuns așteptat
Cum putem identifica o zonă a matricei?	Prin coordonatele a două colțuri opuse (de exemplu, colțul stânga-sus și colțul-dreapta-jos).
Ce subprobleme identificați în problema dată?	<ul style="list-style-type: none"> – citirea datelor de intrare; – verificarea faptului că zona este constituită numai din elemente având aceeași valoare; – determinarea dimensiunii compresiei pentru o zonă dată.
Care dintre aceste subprobleme se rezolvă în mod natural printr-o funcție recursivă? Justificați răspunsul!	<p>Determinarea dimensiunii compresiei unei zone poate fi efectuată astfel:</p> <ul style="list-style-type: none"> – mai întâi verificăm dacă zona este constituită din aceeași valoare (dacă da, returnăm valoarea 2, acesta fiind cazul elementar); – în cazul în care zona nu este constituită din aceeași valoare, împărțim zona în cele 4 subzone (A, B, C, D), determinăm în același mod dimensiunea fiecărei subzone și returnăm ca rezultat $1 + \text{suma dimensiunilor subzonelor}$. <p>Observăm că rezolvarea subproblemei se reduce la rezolvare a 4 subprobleme de același tip, rezolvate în același mod, deci o abordare recursivă este naturală.</p>

Este eficientă o abordare recursivă? Justificați răspunsul!	Abordarea recursivă este eficientă, subproblemele sunt independente (nu se suprapun).
Indicați pe imagine care sunt coordonatele colțurilor subzonelor A, B, C, D în ipoteza că zona I are colțul stânga-sus (x_s, y_s) , respectiv colțul dreapta-jos (x_d, y_d) .	

Etapa 2. Implementare

Scrieți un program care să implementeze această abordare! Trimiteți sursa pe arhiva educațională .campion pentru a fi evaluată automat!

```
#include <fstream>
#define NMax 252
using namespace std;
ifstream fin("imagine.in");
ofstream fout("imagine.out");
int a[NMax][NMax];
int L;
void Citire();
bool Verif(int xs, int ys, int xd, int yd);
int Dim(int xs, int ys, int xd, int yd);
int main ()
{ Citire();
  fout<<Dim(1,1,L,L)<<'\n';
  fout.close();
  return 0;
}

void Citire()
{int i, j;
  fin>>L;
  for (i=1; i<=L; i++)
    for (j=1; j<=L; j++)
      fin>>a[i][j];
}

bool Verif(int xs, int ys, int xd, int yd)
{int i, j, v=a[xs][ys];
  for (i=xs; i<=xd; i++)
    for (j=ys; j<=yd; j++)
      if (a[i][j]!=v) return 0;
  return 1;
}
```

```

int Dim(int xs, int ys, int xd, int yd)
{if (Verif(xs,ys,xd,yd)) return 2;
  return 1+Dim(xs,ys, (xs+xd)/2, (ys+yd)/2)+ //B
        Dim((xs+xd)/2+1,ys,xd, (ys+yd)/2)+ //A
        Dim((xs+xd)/2+1, (ys+yd)/2+1, xd, yd)+//C
        Dim(xs, (ys+yd)/2+1, (xs+xd)/2, yd); //D
}

```

Notă pentru profesori:

Dacă vom trimite pe arhiva educațională .campion această soluție, vom obține borderoul de evaluare:

Test	Timp	Mesaj evaluator	Punctaj
0	0	Okay!	10
1	0	Okay!	10
Total exemple			20
2	0	Okay!	10
3	0	Okay!	10
4	0	Okay!	10
5	0	Okay!	10
6	0	Okay!	10
7	0	Okay!	10
8	0	Okay!	10
9	0.001	Okay!	10
10	0.002	Okay!	10
11	0.003	Okay!	10
Total evaluare			100

Pentru diferite abordări se pot obține punctaje parțiale. De exemplu, un elev care greșește unul dintre apeluri (pentru subzona B), astfel:

```

int Dim(int xs, int ys, int xd, int yd)
{if (Verif(xs,ys,xd,yd)) return 2;
  return 1+Dim(xs,ys, (xs+ys)/2, (ys+yd)/2)+//B
        Dim((xs+xd)/2+1,ys,xd, (ys+yd)/2)+//A
        Dim((xs+xd)/2+1, (ys+yd)/2+1, xd, yd)+//C
        Dim(xs, (ys+yd)/2+1, (xs+xd)/2, yd); //D
}

```

va obține un punctaj parțial după cum este vizibil în borderoul de evaluare următor:

Test	Timp	Mesaj evaluator	Punctaj
0	0	Killed by signal: 11	0
1	0	Okay!	10
Total exemple			10
2	0	Okay!	10
3	0	Killed by signal: 11	0
4	0	Killed by signal: 11	0
5	0	Killed by signal: 11	0
6	0	Okay!	10
7	0	Killed by signal: 11	0
8	0	Okay!	10
9	0.001	Okay!	10
10	0.002	Killed by signal: 11	0
11	0.003	Killed by signal: 11	0
Total evaluare			40

Mesajul `Killed by signal 11` corespunde erorii *Segmentation Fault* (acces ilegal la memorie). Pe anumite seturi de date de test, datorită determinării greșite a colțurilor subzonei B se ajunge la o recursie infinită, ceea ce va genera depășirea memoriei alocate pentru stivă.

Elevii vor fi încurajați să utilizeze facilitățile de *Debug* ale mediului de programare, pentru a identifica erorile din sursa proprie.

Este recomandat ca la sfârșitul activității să propunem spre rezolvare elevilor cel puțin o problemă similară. În funcție de nivelul elevilor, sarcinile de lucru pot fi individualizate. Iată, spre exemplu, exemple de sarcini de lucru similare, dar de niveluri diferite de dificultate:

- exemplul 1 (ușor): problema *ExistaImpareDivImp* (1148) de pe Pbinfo (pbinfo.ro)
- exemplul 2 (mediu): problema *Hambar* (1972) de pe Pbinfo (pbinfo.ro)
- exemplul 3 (difícil): problema *Compresie* de pe arhiva educațională .campion (campion.edu.ro/arhiva)

Activitatea remedială „FSTERGEDuplicate”

Item de evaluare inițială. Exemplul 10 – Nivel 3

Competența specifică:

X.1.2 Prelucrarea datelor structurate

FIȘĂ DE LUCRU

Etapa 1. Analiza

Problema constă în parcurgerea unei liste simplu înlănțuite, identificarea duplicatelor și eliminarea acestora.

Întrebare	Răspuns așteptat
Pentru parcurgerea listei este necesar un pointer auxiliar <code>p</code> . Declarați și inițializați acest pointer!	<code>Nod * p=L;</code>
Există cazuri speciale, care nu necesită o parcurgere a listei? Scrieți o expresie care să fie adevărată dacă și numai dacă lista indicată de pointerul <code>L</code> este caz special.	Lista vidă sau lista formată dintr-un singur nod. <code>!L !L->next</code>
Să considerăm că <code>p</code> indică un nod oarecare al listei, diferit de ultimul nod. Scrieți o expresie care să fie adevărată dacă și numai dacă valoarea memorată în nodul indicat de <code>p</code> coincide cu valoarea memorată în nodul care urmează după cel indicat de <code>p</code> (este o valoare duplicat).	<code>p->val==p->next->val</code>
Presupunând că <code>p</code> indică un nod care conține o valoare duplicat, scrieți o secvență de instrucțiuni care să elimine din listă nodul care urmează după cel indicat de pointerul <code>p</code> . Eliberați zona de memorie alocată nodului eliminat.	<code>Nod * q;</code> <code>q=p->next;</code> <code>p->next=q->next;</code> <code>delete q;</code>
Ce situație specială poate apărea atunci când eliminăm un duplicat? Cum tratați acest caz?	Nodul care urmează după duplicatul eliminat poate să fie de asemenea un duplicat. Pentru a trata această situație nu vom avansa în listă decât nodul curent nu este urmat de un duplicat.

Etapa 2. Implementare

Scrieți definiția funcției `FSTERGEDuplicate()`, în care parcurgeți lista, identificați duplicatele și eliminați aceste duplicate. Trimiteți definiția funcției pe platforma pbinfo, pentru a fi evaluată automat.

```
void FSTERGEDuplicate(Nod *&L)
{Nod *p, *q;
  if (!L || !L->next) return; //lista este vida sau are un singur nod
  p=L;
```

```

while (p->next)
    if (p->val==p->next->val)
    {
        q=p->next;
        p->next=q->next;
        delete q;
    }
    else
        p=p->next;
}

```

Etapa 3.

Urmăriți pas cu pas execuția funcției pentru lista indicată în enunțul problemei, ilustrând la fiecare pas conținutul acesteia.

Notă pentru profesori:

Când această funcție este transmisă pe platforma Pbinfo se va obține următorul borderou de evaluare:

Test	Timp	Mesaj evaluare	Scor posibil	Scor obținut
0	0	OK	10	10
1	0	OK	10	10
2	0	OK	10	10
3	0	OK	10	10
4	0	OK	10	10
5	0	OK	10	10
6	0	OK	10	10
7	0	OK	10	10
8	0	OK	10	10
9	0	OK	10	10

Rezultat final: 100 puncte

3.2. EXEMPLE DE ACTIVITĂȚI DE RECUPERARE

1. Activitatea de învățare „Ghicește numărul”

Competențe specifice

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi

X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

X. 4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

XI.2.1 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei

XI.2.2 Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare, sau a unor probleme cu aplicabilitate practică

Notă pentru profesori:

Recomandăm dezvoltarea competențelor specifice prin aplicații de sinteză pentru unitatea de învățare „Metode de programare” din programa clasei a XI-a.

Activitatea are mai multe sarcini de lucru, gradate ca nivel de dificultate: primele două sarcini de lucru sunt de nivel 1, următoarea este de nivel 2, iar ultima sarcină de lucru este de nivel 3.

1. Ghicește tu numărul – Nivel 1

Tu și calculatorul ați putea juca împreună următorul joc: calculatorul „se gândește” la un număr cuprins între 1 și 1000, iar tu încerci să-l ghicești, evident punând cât mai puține întrebări.

La o întrebare tu menționezi un singur număr natural cuprins în intervalul $[1, 1000]$. Calculatorul îți va răspunde, în funcție de caz astfel:

- „Bravo, ai ghicit!”
- „Numărul este prea mai mare!”
- „Numărul este prea mai mic!”

În ipoteza că numărul ales de calculator este 13, care ar fi o secvență cu număr minim de întrebări prin care ai putea ghici numărul?

Experimentați acest joc folosind una dintre următoarele legături:

- pe site-ul Math Is Fun https://www.mathsisfun.com/games/guess_number.html
- pe site-ul Fun Brain <https://www.funbrain.com/games/guess-the-number>
- pe site-ul Khan Academy <https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/a-guessing-game>

Notă pentru profesori:

Aceste aplicații sunt în limba engleză, prin urmare, dacă este necesar, profesorul va traduce elevilor interacțiunea cu aplicația într-un prim joc demonstrativ. În jocul demonstrativ profesorul nu va juca optim, pentru a nu sugera elevilor strategia de joc. Elevii vor utiliza aplicația, până când vor identifica strategia optimă de joc. O altă posibilitate ar fi ca profesorul să creeze o aplicație în care calculatorul "alege" un număr și elevul trebuie ghicește numărul respectiv folosind un număr minim de întrebări de tipul din enunț.

2: Formularea algoritmului aplicat în limbaj natural – Nivel 1

Întrebare	Răspuns așteptat
Formulați o subproblemă a problemei date	Determinați numărul căutat, știind că acest număr aparține intervalului $[a, b]$.
Care este numărul specificat în prima întrebare? Cum se calculează acesta?	Prima întrebare pe care o punem constă în mijlocul acestui interval $m_{ij} = (a+b) / 2$.
Cum procedați dacă răspunsul la întrebare este: „Bravo, ai ghicit!”	Jocul se termină
Cum procedați dacă răspunsul la întrebare este: „Numărul este prea mai mare!”?	În acest caz jocul va continua, în același mod pentru intervalul $[a, m_{ij}-1]$.
Cum procedați dacă răspunsul la întrebare este: „Numărul este prea mai mic!”?	În acest caz jocul va continua, în același mod pentru intervalul $[m_{ij}+1, b]$.

3. Implementarea aplicației "Calculatorul ghicește numărul!" – Nivel 2

Implementați o aplicație în care calculatorul să ghicească numărul ales de utilizator.

Notă pentru profesori:

Profesorul va oferi elevilor, dacă este necesar, câteva "puncte de sprijin":

1. Observați că pentru a ghici numărul, inițial îl "căutăm" în intervalul $[a, b]$, dar apoi, după o singură întrebare, reducem spațiul de căutare la intervalul $[a, m_{ij}-1]$ sau $[m_{ij}+1, b]$ (unde $m_{ij} = (a+b) / 2$ este mijlocul intervalului) și conținuăm căutarea **în același mod**, deci o implementare recursivă ar fi naturală. Ce parametri va avea funcția recursivă? Declarați funcția!
2. Stabiliți un mod simplu de interacțiune cu calculatorul. De exemplu, puteți codifica răspunsul
 - „Bravo, ai ghicit!” cu semnul =;
 - „Numărul este prea mai mare!” cu semnul >;
 - „Numărul este prea mai mic!” cu semnul <.

4. Analiza complexității – Nivel 3

Determinați matematic numărul minim de întrebări necesare pentru a ghici numărul!

2. Activitatea de învățare „Fractali”

Competențe specifice

X.3.2 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme

X.3.3 Analiza problemei în scopul identificării subproblemelor acesteia

X.3.4 Descrierea metodei de rezolvare a unei probleme în termeni recursivi

X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

XI.2.1 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei

XI.2.1 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei

XI.2.2 Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare, sau a unor probleme cu aplicabilitate practică

Notă pentru profesori:

Recomandăm dezvoltarea competențelor specifice prin aplicații de sinteză pentru unitatea de învățare „Metode de programare” din programa clasei a XI-a.

Activitatea are mai multe sarcini de lucru, gradate ca nivel de dificultate: primele două sarcini de lucru sunt de nivel 1, următoarele trei sunt de nivel 2, iar ultima sarcină de lucru este de nivel 3.

Modul de organizare a activității la clasă este la decizia profesorului, în funcție de specificul clasei. O posibilitate de abordare ar fi ca primele 2 sarcini de lucru să fie rezolvate de către elevi acasă, după o introducere realizată în clasă de către profesor. Documentul Word realizat de către fiecare elev la cerința 2 va fi transmis de către acesta pe platforma utilizată pentru organizarea activității la clasă (de exemplu, Google Classroom, Microsoft Teams, etc). În clasă, profesorul va solicita unul, doi, trei elevi să prezinte documentul creat și va iniția o discuție în care să solicite părerea elevilor despre filmele vizionate, cu evidențierea structurilor fractale.

A doua etapă constă în rezolvarea cerinței 3 (generarea fractalului). Este recomandat ca această aplicație să fie rezolvată integral în clasă. Sugestii de abordare:

1. Profesorul prezintă pe tablă pas cu pas modul de generare a fractalului și inițiază o conversație euristică prin care să dirijeze elevii spre algoritmul de generare a fractalului. De exemplu:
 - a. Analizând modul în care am construit pas cu pas fractalul, identificați ce operații se repetă!
 - b. Formulați în limbaj natural, în termeni recursivi, metoda aplicată pentru construirea fractalului.
 - c. Identificați secvența pașilor care ar trebui executați în funcția recursivă.
 - d. Ce parametri ar trebui să aibă funcția recursivă?
 - e. Care este condiția de ieșire din recursie?
2. Dacă este necesar, profesorul va introduce/recapitula și câteva elemente de grafică, necesare implementării programului: inițializarea modului grafic, lucrul cu culori, desenarea unui pătrat (interior/contur), determinarea dimensiunilor ecranului, etc.
3. Fiecare elev va implementa independent aplicația. Profesorul va oferi indicații, pe parcursul dezvoltării aplicației, dacă este necesar. Activitatea se va finaliza cu prezentarea unor variante de implementare realizate de către elevi. În cazul în care niciuna dintre implementări nu satisface cerințele, profesorul va construi pas cu pas de la zero codul care implementează cerința problemei, explicând la fiecare pas și evidențiind bug-urile identificate în aplicațiile elevilor sau va realiza acest lucru plecând de la una dintre aplicațiile realizate de către elevi.
4. Prima cerință de modificare a aplicației are rolul de a evidenția mai bine structura fractalului. Pentru a rezolva a doua cerință de modificare însă este necesară o bună înțelegere a modului de funcționare a recursiei și identificarea necesității de a introduce un parametru suplimentar al funcției, utilizarea unei variabile globale nefiind o soluție în acest caz. Se pot analiza comparativ ambele variante de abordare (parametru/variabilă globală), urmărind efectul pe ecran și solicitând elevilor să justifice de ce prin reținerea nivelului curent într-o variabilă globală se obține modelul respectiv.

A treia etapă a activității constă în prezentarea și implementarea unui fractal, la alegerea elevilor. Această activitate poate fi desfășurată individual, dar ar fi recomandat ca elevii să lucreze în perechi. Prezentarea Powerpoint, precum și codul sursă al aplicației vor fi transmise de către elevi pe platforma de organizare a activității la clasă. Profesorul va oferi feed-back pentru fiecare prezentare/aplicație și va oferi elevilor posibilitatea de a valorifica feed-back-ul primit, de a îmbunătăți aplicația și de a retrimite materialele pe platformă. La clasă fiecare pereche va prezenta aplicația creată, prezentările fiind urmate de o scurtă sesiune de întrebări, opinii, aprecieri ale colegilor referitoare la aplicația prezentată.

Sarcini de lucru

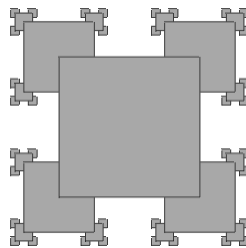
Istoria fractalilor a început brusc în 1975, când matematicianul *Benoit Mandelbrot* a publicat o lucrare revoluționară – „*O teorie a seriilor fractale*”, urmată de o carte de referință „*Geometria fractală a naturii*”. Primii fractali au fost creați de matematicieni ca *Waclaw Sierpinski*, *David Hilbert*, *Georg Cantor*, sub forma unor exerciții abstracte, fără ca aceștia să aibă inițial idee de semnificația lor.

Informal, fractalii pot fi descriși ca forme pentru care fiecare parte se aseamănă cu întregul.

Definiție

Fractalul reprezintă o figură geometrică fragmentată sau frântă care poate fi divizată în părți, astfel încât fiecare dintre acestea să fie o copie miniaturală a întregului.

- Urmăriți pe Youtube câteva filme în care sunt ilustrați fractali. De exemplu:
 - How fractals can help you understand the universe | BBC Ideas
 - Fractals in Nature
 - Fractalii și arta dezordinii (Benoît Mandelbrot la TED)
 - Fractali. Imagini fractale
- Căutați pe Internet câteva modele fractale și inserați într-un document Word câte o imagine cu fractalul respectiv:
 - triunghiul lui Sierpinski
 - curba lui Koch
 - mulțimea lui Mandelbrot
 - cel puțin trei exemple de fractali care se regăsesc în natură
 - cel puțin trei exemple de fractali pe care le regăsiți în artă (indicați sursa).
- Să generăm și noi un fractal! În figura de mai jos este reprezentat un fractal simplu, obținut prin iterarea unui pătrat:



Scrieți un program recursiv care să afișeze pe ecran în mod grafic un astfel de fractal. Latura pătratului central L se citește de la tastatură ($0 < L < 260$); centrul acestui pătrat va coincide cu centrul ecranului. Iterarea se face până când se obțin pătrate de latură 1.

- Modificați programul astfel încât pătratele să aibă interiorul de culoare roșie și conturul de culoare galbenă.
- Modificați din nou programul astfel încât culoarea interioarelor să alterneze pe niveluri (primul pătrat să aibă interiorul de culoare roșie, cele 4 pătrate cu centrele în vârfurile primului pătrat să aibă interioarele albastre, apoi pătratele cu centrele în vârfurile celor 4 pătrate de pe al doilea nivel vor avea din nou interioarele roșii, ș.a.m.d.
- Alegeți un model fractal. Realizați o prezentare Powerpoint în care să descrieți fractalul respectiv, cu explicitarea suportului matematic. Implementați un program care să genereze pe ecran în mod grafic fractalul, pas cu pas. Includeți în prezentarea Powerpoint capturi de ecran cu fractalul generat de programul vostru.

Evaluarea activității

Ponderea punctajelor în evaluarea activității

Cerința	2	3+4+5	6
Pondere	10%	50%	40%

Cerința 2:

Se acordă câte 1 punct pentru fiecare imagine cu fractali conform cerinței din documentul Word și 1 punct din oficiu.

Cerințele 3-4-5

Pentru fiecare criteriu sunt prezentați descriptori pentru 4 niveluri de performanță, modul de alocare a punctajelor fiind la decizia profesorului.

Criteriu	10-9	8-7	6-5	4-3-2-1
1. Proiectarea funcției recursive	Antetul funcției este corect. Funcția tratează cazul elementar Funcția conține toate cele 4 apeluri recursive corecte. Funcția generează fractalul specificat.	Antetul funcției este corect. Funcția tratează cazul elementar Funcția conține cel puțin 3 dintre cele 4 apeluri recursive corecte. Fractalul generat de funcție nu este identic cu fractalul specificat.	Antetul funcției este corect. Funcția recursivă este principial corectă, dar nu funcționează deoarece fie nu este tratat cazul elementar, fie nu sunt toate cele 4 apeluri recursive corecte.	Funcția recursivă este incorectă.
2. Elementele de grafică	Elementele de grafică specificate sunt corecte: <ul style="list-style-type: none"> – modul grafic este inițializat corect – se utilizează pătrate cu interior roșu – centrul pătratului inițial coincide cu centrul ecranului – centrele celorlalte pătrate sunt poziționate corect – dimensiunile laturilor sunt corecte – după modificarea de la cerința 4 conturul este galben 	Elementele de grafică specificate sunt în mare majoritate corecte (una sau două cerințe nu sunt respectate)	Elementele de grafică specificate sunt parțial corecte	Niciuna sau cel mult două dintre cerințele grafice sunt respectate
3. Alternanța culorilor	După modificarea de la cerința 5, culorile alternează conform enunțului	Elevul identifică nivelurile în mod corect, dar culorile nu alternează corect.	Culorile alternează, dar nu pe niveluri	Culorile nu alternează.

Cerința 6

A. Realizarea aplicației și a prezentării PowerPoint (70% din punctaj)

Criteriu	10-9	8-7	6-5	4-3-2-1
1. Prezentarea Powerpoint	1. are un conținut corect și adecvat (descrierea fractalului, prezentarea suportului	Are un conținut corect și adecvat dar, structura/	Are un conținut parțial corect sau incomplet.	Are un conținut sumar puțin

	matematic care va conduce la generarea fractalului, imagini cu fractalul respectiv, capturi de ecran din aplicația creată de elev; sunt menționate resursele folosite); 2. are structură și formatare corecte și atractive.	formatarea nu respectă principiile generale ale unei prezentări PowerPoint.		sau deloc relevant.
2. Funcționalitate	Aplicația este funcțională, generează fractalul prezentat. Există elemente de parametrizare a fractalului	Aplicația este în funcțională, dar generează corect un singur model fractal.	Aplicația este principial corectă, dar conține bug-uri care nu permit generarea fractalului	Aplicația nu există sau este incorectă.
3. Complexitate tehnică	Aplicația este complexă, necesită abilități de analiză și programare de nivel relativ înalt	Aplicația are complexitate medie din punctul de vedere al analizei, dar necesită bune abilități de programare	Aplicația are complexitate medie	Aplicația nu există sau ilustrează competențe de programare la nivel elementar.

B. Prezentarea proiectului (20% din punctaj)

Criteria	10-9	8-7	6-5	4-3-2-1
1. Cunoașterea conținutului proiectului	Elevii cunosc complet conținutul proiectului.	Elevii cunosc în mare proporție conținutul proiectului	Elevii cunosc în oarecare măsură conținutul proiectului	Elevii cunosc foarte puțin din conținutul proiectului
2. Voce	Fiecare cuvânt rostit poate fi auzit clar și fără dificultăți de audiență Vorbitorul utilizează un volum adecvat al vocii, un ton adecvat, are inflexiuni ale vocii, gesturi și expresii faciale care să atragă atenția auditoriului asupra proiectului. Prezentarea este convingătoare, denotă energie și entuziasm.	Fiecare cuvânt rostit poate fi auzit clar și fără dificultăți de audiență. Vorbitorul face eforturi să fie clar, expresiv și să atragă atenția audienței.	Cea mai mare parte a prezentării poate fi auzită clar și fără dificultăți de audiență. Tonul și gesturile sunt inexpresive, plictisitoare.	Prezentatorul vorbește încet și neclar. Tonul este monoton. Lipsește energia și entuziasmul.
3. Ritm	Ritmul prezentării este natural, adecvat (nici prea lent pentru a fi plictisitor, nici prea alert pentru a nu putea fi înțeles conținutul proiectului). Nu există pauze inadecvate.	Ritmul prezentării este puțin prea lent sau puțin prea rapid.	Ritmul prezentării este prea lent sau prea rapid, apar pauze nejustificate în prezentare.	Prezentare nu are ritm adecvat, apar numeroase pauze în prezentare.
4. Contact vizual	Contactul vizual cu audiența este permanent	Nu este realizat permanent contactul vizual cu audiența	Contactul vizual cu audiența este slab	Nu este realizat un contact vizual cu audiența

5. Exersare	Este evident faptul că prezentarea a fost exersată, vorbitorii se sincronizează perfect. Nu sunt utilizate notițe sau materiale ajutătoare.	Este evident faptul că prezentarea a fost exersată, vorbitorii cunosc structura prezentării. Nu sunt utilizate notițe sau materiale ajutătoare.	Este evident faptul că prezentarea nu a fost exersată suficient, vorbitorii nu se sincronizează, sunt utilizate notițe sau materiale ajutătoare.	Prezentarea nu a fost exersată
6. Lungimea prezentării	Prezentarea a fost realizată în timpul alocat	Prezentarea este puțin mai lungă/scurtă decât timpul alocat	Prezentarea este semnificativ mai lungă/scurtă față de timpul alocat	Prezentarea este mult prea lungă/scurtă față de timpul alocat

C. Evaluarea și autoevaluarea (10% din punctaj)

Criteria	10-9	8-7	6-5	4-3-2-1
Evaluare/ autoevaluare	Pentru fiecare criteriu de evaluare notele sunt acordate în mod adecvat și sunt însoțite de justificări	Pentru majoritatea criteriilor de evaluare notele sunt acordate în mod adecvat și sunt însoțite de justificări	În general notele sunt acordate în mod adecvat	În cea mai mare parte notele nu sunt acordate adecvat și nu există justificări.

Notă pentru profesori:

La adresa <http://ler.is.edu.ro/~ema/fractali.html> sunt publicate câteva exemplificări de prezentare Powerpoint, realizate de elevi de clasa a X-a.

3. Activitatea de învățare „Parcurgere DFS” – Nivel 2

Competențe specifice:

X.1.1 Evidențierea necesității structurării datelor

X.1.2 Prelucrarea datelor structurate

X.1.3 Alegerea structurii de date adecvate rezolvării unei probleme

XI.1.2 Analizarea unei probleme în scopul identificării datelor necesare și alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă

X.2.3 Aplicarea mecanismului recursivității prin crearea unor subprograme recursive (definite de utilizator)

XI.1.4 Descrierea algoritmilor fundamentali de prelucrare a grafurilor și implementarea acestora într-un limbaj de programare

XI.1.5 Descrierea operațiilor specifice listelor simplu înlănțuite și elaborarea unor subprograme care să implementeze aceste operații

XI.1.7 Analizarea în mod comparativ a avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme

Notă pentru profesori:

Recomandăm această activitate pentru dezvoltarea competențelor specifice menționate în cadrul unității de învățare „Grafuri neorientate și grafuri orientate” din programa clasei a XI-a.

FIȘĂ DE LUCRU

Să se rezolve problema "Parcurgere DFS – componente conexe" din arhiva educațională Infoarena (<https://infoarena.ro/problema/dfs>)

Se dă un graf neorientat cu N noduri și M muchii.

Cerință

Să se determine numărul componentelor conexe ale grafului.

Date de intrare

Fișierul de intrare `dfs.in` conține pe prima linie numerele naturale N și M cu semnificația din enunț, Pe următoarele M linii se găsesc câte două numere X și Y cu semnificația: există muchie între nodurile X și Y .

Date de ieșire

Fișierul de ieșire `dfs.out` va conține o singură linie pe care va fi scris numărul de componente conexe ale grafului.

Restricții

- $1 \leq N \leq 100000$
- $0 \leq M \leq \min(200000, N * (N+1) / 2)$
- Pentru 50% dintre teste $1 \leq N \leq 1000$

Exemplu

dfs.in	dfs.out
6 3	3
1 2	
1 4	
3 5	

Abordare posibilă

Soluția problemei este indicată în titlul acesteia. Apelând funcția `dfs(x)` parcurgem în adâncime toate vârfurile din componenta conexă a vârfului x . În timpul parcurgerii, vom marca vârfurile vizitate în vectorul `uz` (`uz[i]=1`, dacă vârful i a fost vizitat, respectiv `0` dacă vârful i nu a fost vizitat).

În funcția `main()` vom parcurge vectorul `uz` și când identificăm un vârf nevizitat, deducem că am descoperit o nouă componentă conexă (o numărăm în variabila `nrc` și apelăm funcția `dfs` din vârful respectiv, pentru a vizita toate vârfurile din componenta conexă a acestuia).

Observație

Puteți reactualiza cunoștințele despre parcurgerea DFS utilizând aplicația educațională *Teoria grafurilor* de pe arhiva educațională `.campion` (secțiunea *Software*).

http://campion.edu.ro/arhiva/www/arhiva_2009/seds/17/index.htm

Reprezentarea informațiilor

Vom reprezenta graful prin liste de adiacență, adică vom utiliza un vector cu n componente, în care vom reține pentru fiecare vârf din graf lista sa de adiacență.

Lista de adiacență a vârfului x va conține toate vârfurile y cu proprietatea că există muchia $[x, y]$ (pentru graf neorientat). Ordinea în care sunt memorate vârfurile într-o listă de adiacență nu contează.

Am ales această reprezentare, deoarece $1 \leq N \leq 100000$, prin urmare nu avem suficientă memorie necesară pentru a reprezenta matricea de adiacență.

În plus, parcurgerea DFS utilizând reprezentare prin matrice de adiacență are complexitatea timp $O(N^2)$, prea mare pentru timpul de execuție specificat.

Utilizând liste de adiacență, complexitatea timp a parcurgerii DFS este $O(N+M)$. Memoria necesară depinde de modul de implementare a listelor de adiacență.

Varianta 1:

O listă de adiacență poate fi reprezentată ca un vector, în care vârfurile sunt memorate pe poziții consecutive. Observăm că numărul de elemente din lista de adiacență diferă de la un vârf la altul, fiind egal cu gradul, respectiv gradul exterior al vârfului, grad cuprins între 0 și $n-1$ (n fiind numărul de vârfuri din graf). În primul rând va fi necesar să memorăm gradul, respectiv gradul exterior al fiecărui vârf. Acest lucru se poate realiza într-un alt vector (vectorul gradelor) sau îl putem memora pe poziția 0 a vectorului în care memorăm lista de adiacență.

Declarații:

```
#define Nmax 1001
int n, A[Nmax][Nmax];
```

Gradul vârfului x va fi memorat în $A[x][0]$, vârfurile adiacente cu x fiind $A[x][1]$, $A[x][2]$, ..., $A[x][A[x][0]]$.

Implementând soluția problemei utilizând această reprezentare, vom obține doar 50 de puncte (50% din punctaj, pentru $1 \leq N \leq 1000$).

```
#include <fstream>
#define NMAX 1001

using namespace std;

ifstream fin("dfs.in");
ofstream fout("dfs.out");
int n;
int A[NMAX][NMAX];
int uz[NMAX];
int nrc;

void citire();
void dfs(int x);

int main()
{int i;
  citire();
  for (i=1; i<=n; i++)
    if (!uz[i])
      {
        nrc++;
        dfs(i);
      }
  fout<<nrc<<'\n';
  return 0;
}

void citire()
{int i, x, y, m;
  fin>>n>>m;
  for (i=0; i<m; i++)
    {
      fin>>x>>y;
      A[x][0]++; A[x][A[x][0]]=y;
      A[y][0]++; A[y][A[y][0]]=x;
    }
}

void dfs(int x)
{int i;
  uz[x]=1;
  for (i=1; i<=A[x][0]; i++)
    if (!uz[A[x][i]])
      dfs(A[x][i]);
}
```


Compilare:

Test	Timp executie	Memorie folosita	Mesaj	Punctaj/test
1	3ms	262kb	OK	5
2	3ms	524kb	OK	5
3	3ms	131kb	OK	5
4	13ms	917kb	OK	5
5	4ms	786kb	OK	5
6	7ms	1572kb	OK	5
7	9ms	2490kb	OK	5
8	13ms	3276kb	OK	5
9	13ms	3801kb	OK	5
10	14ms	4194kb	OK	5
11	2ms	131kb	Killed by Signal 11	0
12	3ms	131kb	Killed by Signal 11	0
13	2ms	131kb	Killed by Signal 11	0
14	2ms	131kb	Killed by Signal 11	0
15	2ms	131kb	Killed by Signal 11	0
16	2ms	131kb	Killed by Signal 11	0
17	14ms	4194kb	Killed by Signal 11	0
18	2ms	131kb	Killed by Signal 11	0
19	2ms	131kb	Killed by Signal 11	0
20	2ms	131kb	Killed by Signal 11	0
Punctaj total				50

Varianta 2:

Vom reprezenta o listă de adiacență ca o listă simplu înlănțuită, în care vom reține nodurile din lista de adiacență.

Declarații:

```
#define NMAX 100001
struct Nod
{
    int vf;
    struct Nod * urm;
};
typedef struct Nod * Lista;
int n;
Lista A[NMAX];
```

Funcțiile `citire()` și `dfs()` se vor modifica pentru a utiliza această reprezentare astfel:

```
void citire()
{
    int i, x, y, m;
    Lista p;
    fin>>n>>m;
    for (i=0; i<m; i++)
    {
        fin>>x>>y;
        //inseram pe x in lista lui y
        p=new Nod; p->vf=x; p->urm=A[y]; A[y]=p;
        //inseram pe y in lista lui x
        p=new Nod; p->vf=y; p->urm=A[x]; A[x]=p;
    }
}
```

```
void dfs(int x)
{
    Lista p;
    uz[x]=1;
    for (p=A[x]; p; p=p->urm)
        if (!uz[p->vf])
            dfs(p->vf);
}
```

Această abordare obține 100 de puncte.

Compilare:

Test	Timp executie	Memorie folosita	Mesaj	Punctaj/test
1	2ms	131kb	OK	5
2	2ms	131kb	OK	5
3	2ms	131kb	OK	5
4	16ms	1441kb	OK	5
5	2ms	131kb	OK	5
6	2ms	131kb	OK	5
7	3ms	131kb	OK	5
8	3ms	262kb	OK	5
9	4ms	393kb	OK	5
10	4ms	262kb	OK	5
11	32ms	2621kb	OK	5
12	22ms	1835kb	OK	5
13	31ms	2621kb	OK	5
14	33ms	2752kb	OK	5
15	43ms	3407kb	OK	5
16	55ms	4194kb	OK	5
17	89ms	9306kb	OK	5
18	5ms	524kb	OK	5
19	8ms	1441kb	OK	5
20	62ms	4980kb	OK	5
Punctaj total				100

Ceva nu functioneaza?

Varianta 3:

Vom utiliza biblioteca STL și vom implementa o listă de adiacență utilizând clasa `vector` din STL (în cadrul acestei clase, memoria pentru vector este alocată dinamic, prin urmare ne vom încadra în memoria totală disponibilă).

Declarații

```
#define NMAX 100001
int n;
vector<int> A[NMAX];
```

Funcțiile `citire()` și `dfs()` se vor modifica pentru a utiliza această reprezentare astfel:

```
void citire()
{int i, x, y, m;
  fin>>n>>m;
  for (i=0; i<m; i++)
  {
    fin>>x>>y;
    //inseram pe x in lista lui y
    A[x].push_back(y);
    //inseram pe y in lista lui x
    A[y].push_back(x);
  }
}

void dfs(int x)
{int i;
  uz[x]=1;
  for (i=0; i<A[x].size(); i++)
    if (!uz[A[x][i]])
      dfs(A[x][i]);
}
```

Și această abordare obține 100 de puncte.

Test	Timp execuție	Memorie folosita	Mesaj	Punctaj/test
1	7ms	2490kb	OK	5
2	7ms	2490kb	OK	5
3	7ms	2490kb	OK	5
4	16ms	2752kb	OK	5
5	7ms	2490kb	OK	5
6	7ms	2490kb	OK	5
7	8ms	2621kb	OK	5
8	8ms	2621kb	OK	5
9	9ms	2621kb	OK	5
10	9ms	2621kb	OK	5
11	36ms	3407kb	OK	5
12	30ms	3407kb	OK	5
13	43ms	3801kb	OK	5
14	48ms	3932kb	OK	5
15	61ms	4325kb	OK	5
16	76ms	4849kb	OK	5
17	100ms	9306kb	OK	5
18	9ms	2883kb	OK	5
19	12ms	3145kb	OK	5
20	86ms	4980kb	OK	5
Punctaj total				100

Comparație

Dacă vom analiza borderourile de evaluare pentru ultimele două variante, vom observa că timpii de execuție pentru varianta 2 sunt mai buni (complexitatea teoretică pentru cele două implementări este aceeași, dar lucrul cu clase și obiecte mărește timpul de execuție al programului). De asemenea puteți observa și diferențe în memoria utilizată de program, tot în favoarea variantei 2. Acest lucru se datorează modului în care se alocă dinamic memoria în clasa `vector` (la o nouă realocare se alocă un număr de elemente egal cu cea mai mică putere a lui 2 mai mare sau egală cu dimensiunea necesară). În concluzie, implementarea listelor de adiacență ca liste simplu înlănțuite este optimă.

Exercițiu de același tip

Propunem spre rezolvare un exercițiu similar, bazat de această dată pe parcurgerea BFS.

<https://infoarena.ro/problema/bfs>

Clasa a X-a (Matematică-informatică și Științe ale naturii)

Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar

În vederea elaborării planificării calendaristice se recomandă următorul demers:

- se analizează programele școlare în vigoare, clasele a IX-a și a X-a - disciplina Informatică
 - o programa clasei a IX-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_clasa%20a%20IX-a.pdf
 - o programa clasei a X-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_clasa%20a%20X-a.pdf
- din programa clasei a IX-a se extrag acele competențe specifice a căror structurare ar fi trebuit realizată preponderent în semestrul al II-lea (în perioada în care învățarea s-a realizat la distanță) și conținuturile aferente;
- se analizează competențele specifice din programa clasei a X-a și se stabilesc legături între acestea și competențele nestructurate/ parțial structurate în anul școlar 2019-2020;
- se analizează conținuturile neabordate/parțial abordate din clasa a IX-a și se stabilește modul de fuzionare cu elementele de conținutul aferent clasei a X-a, astfel încât logica internă a domeniului să fie păstrată în condițiile structurării competențelor specifice ale clasei a X-a.

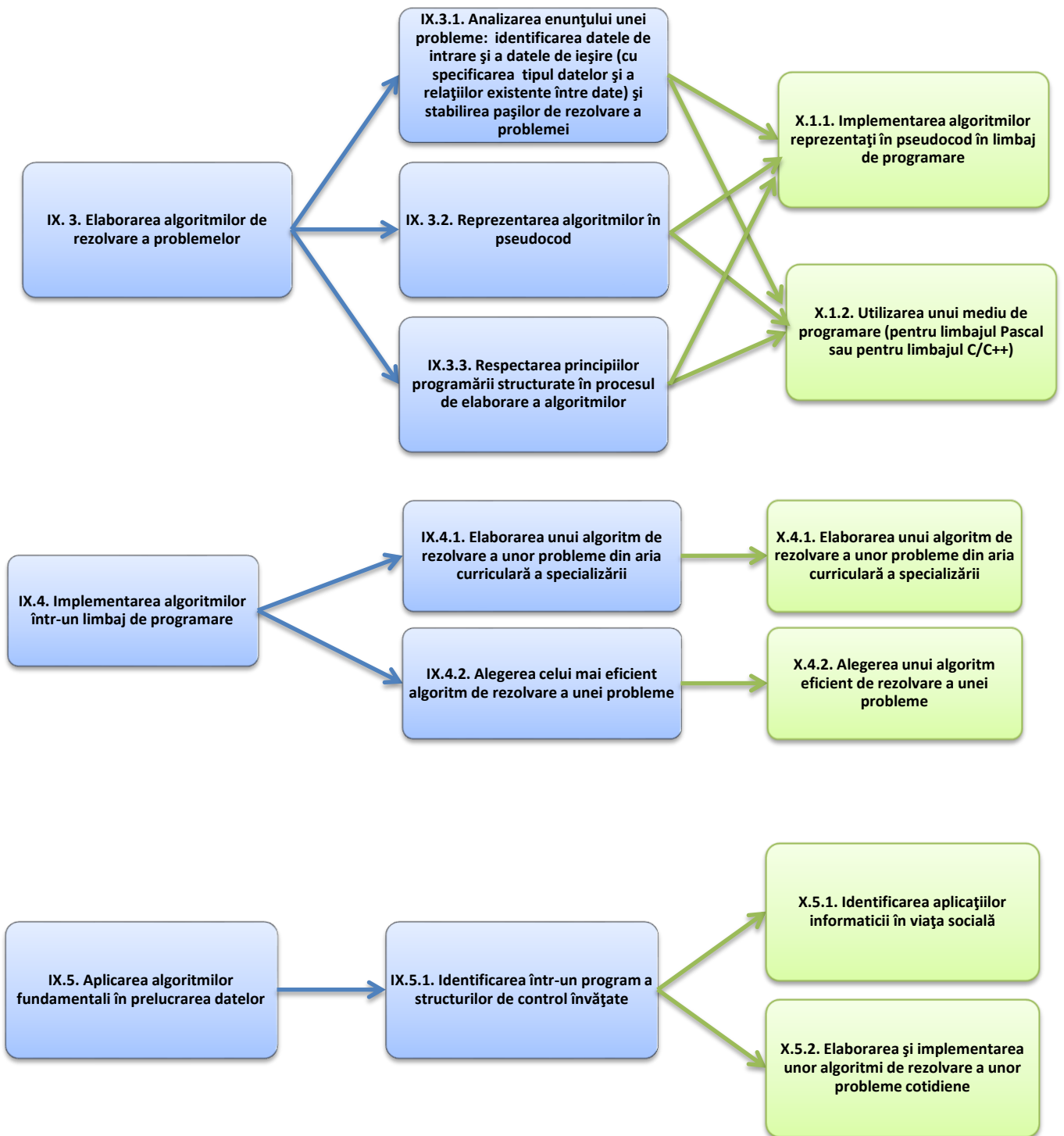
Prin analizarea planificării calendaristice a anului școlar 2019-2020 pentru clasa a IX-a, se pot identifica probleme în ceea ce privește atingerea competențelor IX.3.1, IX.3.2, IX.3.3, IX.4.1, IX.4.2, IX.5.1.

Toate competențele identificate la clasa a IX-a, **Algoritmi elementari** (3.1.Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei, 3.2.Reprezentarea algoritmilor în pseudocod, 3.3.Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor) și **Aplicații interdisciplinare** (4.1.Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării, 4.2.Alegerea celui mai eficient algoritm de rezolvare a unei probleme, 5.1.Identificarea într-un program a structurilor de control învățate) pot fi consolidate în clasa a X-a în **Elemente de bază și mediul de programare al limbajului Pascal sau C/C++**, deoarece toate conținuturile din materia clasei a IX-a pot fi reluate, în mod natural, prin utilizarea unui mediu de programare.

Competența 5.1 se regăsește și în programa clasei a X-a și poate fi dezvoltată în anul școlar 2020-2021 și în cadrul unităților: **Aplicații interdisciplinare și analiza eficienței unui algoritm** respectiv **Aplicații cu algoritmi pentru rezolvarea problemelor cotidiene**.

Pentru atingerea tuturor competențelor, profesorul va prioritiza conținuturile, dirijând învățarea în funcție de gradul de progres al elevilor.

În continuare, sunt prezentate aceste competențe, care sunt corelate cu prevederile programei școlare din clasa a X-a.



Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior

În cadrul acestei secțiuni, exemplificăm acțiunile prin care profesorul face o diagnoză a nivelului achizițiilor elevilor printr-un test de evaluare inițială, cu raportare la următoarele competențe specifice: 2.3, 3.1, 3.2, 3.3, 4.1, 4.2, 5.1.

Item de evaluare inițială. Exemplul 1 – Nivel 1

Competența specifică:

IX.3.2 Reprezentarea algoritmilor în pseudocod

Enunț:

Se consideră următorul algoritm, în care s-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b , cu $[a]$ partea întreagă a numărului real a , descris în pseudocod:

```

citește x (x număr natural)
rezultat ← 0
cât timp x ≠ 0 execută
    dacă x % 2 = 0 atunci
        rezultat ← rezultat + x % 10
    x ← [x / 10]
scrie rezultat

```

Valoarea afișată la finalul acestui algoritm reprezintă:

- suma cifrelor numărului natural x
- produsul cifrelor numărului natural x
- suma cifrelor pare ale numărului natural x
- produsul cifrelor pare ale numărului natural x

Răspuns corect: c

Item de evaluare inițială. Exemplul 2 – Nivel 2

Competența specifică:

IX.3.2 Reprezentarea algoritmilor în pseudocod

Enunț:

Se consideră algoritmul de mai jos, reprezentat în pseudocod. S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b , cu $[a]$ partea întreagă a numărului real a .

```

citește n (număr natural nenul)
p ← 0
pentru i ← 1, n execută
    m ← i
    cât timp m % 2 = 0 execută
        m ← [m / 2]
    dacă m = 1 atunci
        p ← i
scrie p

```

- Ce se va afișa dacă pentru n se citește valoarea 10?
- Dați exemplu de un număr care poate fi citit, astfel încât valoarea afișată să fie 4.

Răspuns corect:

- 8
- Orice număr din mulțimea { 4, 5, 6, 7 }

Item de evaluare inițială. Exemplul 3 – Nivel 3**Competențe specifice:****Item 3.a** IX.3.2 Reprezentarea algoritmilor în pseudocod**Item 3.b** IX.3.2 Reprezentarea algoritmilor în pseudocod**Item 3.c** IX.4.2 Alegerea celui mai eficient algoritm de rezolvare a unei probleme**Enunț:**

Se propun următorii algoritmi descriși incomplet.

```

1)
citește n (n număr natural)
ok ← 1
daca n < 2 atunci
    ok ← 0
    altfel
        pentru i ← 2, [n/2] execută
            dacă ... atunci
                ok ← 0
daca ok = 1 atunci
    scrie n, "este prim"
    altfel
        scrie n, "nu este prim"

```

```

2)
citește n (n număr natural)
ok ← 1
daca n < 2 atunci
    ok ← 0
    altfel
        i ← 2; nr ← 0
        cat timp i*i ≤ n executa
            dacă n%i = 0 atunci
                ok ← 0
            ...
daca ok = 1 atunci
    scrie n, "este prim"
    altfel
        scrie n, "nu este prim"

```

- Completați punctele de suspensie din algoritmul 1) astfel încât acesta să verifice în mod corect primalitatea lui n !
- Completați punctele de suspensie din algoritmul 2) astfel încât acesta să verifice în mod corect primalitatea lui n !
- Care dintre cei doi algoritmi este mai eficient din punct de vedere al timpului de executare?

Răspunsuri corecte:

- $n \% i = 0$
- $i \leftarrow i + 1$
- Algoritmul propus la varianta 2) deoarece are complexitate $O(\sqrt{n})$.

EXEMPLU DE TEST DE EVALUARE INIȚIALĂ**Matricea de specificații:**

Competența specifică	Nivelul 1 (Cunoaștere)	Nivelul 2 (Aplicare)	Nivelul 3 (Raționament)
IX.2.3 Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.	1	2	
IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.	3a	4b	

IX.3.2 Reprezentarea algoritmilor în pseudocod		3b	4c
IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor		3c	4d
IX.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării	4a	3d	

Notă: Timp de lucru 50 de minute.

Itemul 1

Competența specifică:

IX.2.3 Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.

- Variabila întregă n memorează un număr natural cu exact 4 cifre. Identificați atribuirea în urma executării căreia variabila n va memora cifra sutelor din valoarea sa inițială.
 - $n \leftarrow n/1000$
 - $n \leftarrow n/10\%10$
 - $n \leftarrow n\%1000/10$
 - $n \leftarrow n\%1000/100$

Răspuns corect: d

Itemul 2

Competența specifică:

IX.2.3 Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.

- Variabilele a , b și z sunt reale, iar $a \leq b$. Care dintre expresiile următoare are valoarea `true` dacă și numai dacă valoarea variabilei z se află în afara intervalului închis determinat de valorile variabilelor a și b ?
 - $z > a$ sau $z > b$
 - $z < a$ sau $z > b$
 - $z < a$ și $z > b$
 - $z \geq a$ și $z \leq b$

Răspuns corect: b

Itemul 3

Competențe specifice:

Item 3.a IX.3.1. Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei

Item 3.b IX.3.2. Reprezentarea algoritmilor în pseudocod

Item 3.c IX.3.3. Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor

Item 3.d IX.4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

- Se consideră algoritmul alăturat, reprezentat în pseudocod:
 - Ce va afișa dacă pentru variabilele a și b se vor citi valorile 10, respectiv 20?
 - Dați un set de valori de intrare pentru care algoritmul nu afișează nimic.
 - Scrieți un algoritm echivalent care să utilizeze alt tip de structură repetitivă.
 - Dacă a și b reprezintă două zile din calendar și x

```

citește a,b
    (numere naturale, a≤b)
i ← a
cat timp i ≤ b executa
    dacă i%2 = 0 atunci
        scrie i
        i ← i + 1

```

temperatura înregistrată zilnic la o stație meteo, modificați algoritmul pentru a calcula temperatura medie în intervalul calendaristic $[a, b]$.

Răspuns corect 3a: 10 12 14 16 18 20

Răspuns corect 3b: orice valori a, b numere naturale, egale, impare

Răspuns corect 3c: Un răspuns corect poate fi una dintre variantele următoare:

a) citește a, b
(numere naturale, $a \leq b$)
pentru $i \leftarrow a, b$ execută
 dacă $i \% 2 = 0$ atunci
 scrie i

b) citește a, b
(numere naturale, $a \leq b$)
 $i \leftarrow a$
repetă
 dacă $i \% 2 = 0$ atunci
 scrie i
 $i \leftarrow i + 1$
 până când $i > b$

Răspuns corect 3d:

citește a, b (numere naturale, $a \leq b$)

$S \leftarrow 0$

cat timp $i \leq b$ executa
 citeste x
 $S \leftarrow S + x$
 $i \leftarrow i + 1$

scrie S/i

Itemul 4

Competențe specifice:

Item 4.a IX.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Item 4.b IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei

Item 4.c IX.3.2 Reprezentarea algoritmilor în pseudocod

Item 4.d IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor

Gigel se plimbă cu mașina. El parcurge n etape, numerotate de la 1 la n , pentru fiecare etapă cunoscându-se durata și viteza constantă de deplasare. Cu scopul de a calcula distanța totală parcursă și viteza medie de deplasare

- Scrieți expresiile aritmetice de care aveți nevoie pentru a determina distanța totală parcursă și viteza medie de deplasare.
- Identificați care sunt datele de intrare, datele de ieșire și pașii pentru rezolvarea problemei.
- Pe baza pașilor identificați, descrieți în pseudocod algoritmul de rezolvare al problemei.
- Aflați numărul etapei în care viteza de deplasare fost maximă.

Restricții și precizări

Durata se exprimă în ore, distanța în kilometri, iar viteza în km/h.

Exemplu

Date de intrare	Date de ieșire
3	395 65.83
2 65	
3 70	
1 55	

Explicație

În total, Gigel a parcurs $2 \cdot 65 + 3 \cdot 70 + 1 \cdot 55 = 395$ km, în $2 + 3 + 1 = 6$ ore. Viteza medie este $395/6 = 65.83$ km/h

Răspuns corect 4a:

viteza = distanța/timp

viteza_{medie} = distanța totală parcursă/durata totală a deplasării

Răspuns corect 4b:

Date de intrare: numărul natural n , iar apoi n perechi de numere naturale t și v .

Date de ieșire: numerele reprezentând durata totală a deplasării (durata), respectiv viteza medie de deplasare (v_{medie}).

Pas 1: se citește numărul n

Pas 2: se inițializează cu 0 două variabile durata și distanța, în care se vor calcula durata totală a deplasării, respectiv distanța totală parcursă.

Pas 3: se citesc succesiv durata și viteza pentru fiecare dintre cele n intervale și se actualizează valorile durata și distanța

Pas 4: se determină viteza medie

Pas 5: Se afișează datele de ieșire, durata și v_{medie}

Răspuns corect 4c:

citeste n

durata $\leftarrow 0$

distanța $\leftarrow 0$

pentru $i \leftarrow 1, n$ execută

 citeste t, v

 durata \leftarrow durata + t

 distanța \leftarrow distanța + $v \cdot t$

$V_{medie} \leftarrow$ distanța/durata

scrie distanța, v_{medie}

Răspuns corect 4d:

citeste n

max $\leftarrow 0$

pentru $i \leftarrow 1, n$ execută

 citeste t, v

 daca Max < v atunci

 Max $\leftarrow v$

 nr $\leftarrow i$

scrie nr

Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare

3.1. EXEMPLE DE ACTIVITĂȚI REMEDIALE

Activitate remedială „Suma cifrelor”

Item de evaluare inițială. Exemplul 1 – Nivel 1

Competența specifică:

IX.3.2 Reprezentarea algoritmilor în pseudocod

Notă pentru profesori:

Activitatea va fi structurată, având scopul de a construi prin sarcini succesive competența vizată.

Propunem o activitate remedială, în două etape:

- *Etapa 1:* Fișă de lucru pentru ghidarea rezolvării itemului propus.
- *Etapa 2:* Fișă de lucru conținând cerințe suplimentare pentru același enunț.

ETAPA 1. FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Ce valoare va fi afișată dacă pentru variabila x se citește valoarea 724?	$4 + 2 = 6$
Pentru ce valori ale variabilei x , expresia $x \% 2 = 0$ este adevărată?	Numere pare
Dacă $x = 7543$, atunci în urma executării instrucțiunii $x \leftarrow [x/10]$, variabila x va avea valoarea?	754
Pe baza răspunsurilor anterioare, care este valoarea afișată la finalul algoritmului dat?	Răspuns corect c: suma cifrelor pare ale numărului natural x

ETAPA 2. FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Există valori nenule pentru care rezultatul afișat este zero?	Da, numerele care conțin numai cifre impare sau numerele naturale care conțin cifre impare și cifra 0.
Modificați algoritmul astfel încât să se afișeze suma tuturor cifrelor?	Pentru a determina suma tuturor cifrelor, atribuirea rezultat \leftarrow rezultat + $x \% 10$ trebuie executată indiferent de paritatea cifrelor, prin urmare structura de decizie nu mai este necesară. citește x (x număr natural) rezultat \leftarrow 0 cât timp $x \neq 0$ execută rezultat \leftarrow rezultat + $x \% 10$ $x \leftarrow [x/10]$ scrie rezultat
Care sunt modificările necesare astfel încât algoritmul să afișeze produsul cifrelor pare?	citește x (x număr natural) rezultat \leftarrow 1 // se inițializează cu 1 cât timp $x \neq 0$ execută dacă $x \% 2 = 0$ atunci rezultat \leftarrow rezultat * $x \% 10$ // produs $x \leftarrow [x/10]$ scrie rezultat

Activitate remedială „Puterile lui doi”

Item de evaluare inițială. Exemplul 2 – Nivel 2

Competența specifică:

IX.3.2. Reprezentarea algoritmilor în pseudocod

Notă pentru profesori:

Propunem o activitate remedială, în două etape:

- *Etapa 1:* Fișă de lucru pentru ghidarea rezolvării itemului propus.
- *Etapa 2:* Fișă de lucru conținând cerințe suplimentare pentru același enunț.

Etapa 1

FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Urmărind secvența de instrucțiuni de mai jos, în ce condiții se va afișa variabila i ? $m \leftarrow i$ cât timp $m \% 2 = 0$ execută $m \leftarrow [m/2]$ ■ dacă $m = 1$ atunci scrie i ■	Dacă i este putere a lui 2
Pornind de la răspunsul anterior, care este valoarea afișată la finalul algoritmului dat, dacă pentru n se citește valoarea 10?	8
Ce se va afișa pentru $n = 4$?	4
Ce se va afișa pentru $n = 5$?	4
Care este mulțimea valorilor ce pot fi citite pentru n astfel încât să se afișeze 4	Orice număr din mulțimea $\{4, 5, 6, 7\}$

Etapa 2

FIȘĂ DE LUCRU

1. Scrieți cel mai mic și cel mai mare număr care pot fi citite, astfel încât, pentru fiecare dintre acestea, valoarea afișată să fie 16.
2. Formulați un enunț de problemă a cărei rezolvare să poată fi reprezentată de algoritmul dat.

Întrebare	Răspuns așteptat
Ce valoare va fi afișată dacă pentru variabila n se citește valoarea 14?	8
Care este mulțimea valorilor ce pot fi citite pentru n astfel încât să se afișeze 16	16, 17, 18..., 31
Scrieți un enunț de problemă a cărei rezolvare să fie algoritmul dat.	Se citește un număr natural n cu cel mult 9 cifre. Să se determine cea mai mare putere a lui 2 cu proprietatea că este mai mică sau egală cu n .

Activitate remedială „Divizori”

Item de evaluare inițială. Exemplul 3 – Nivel 3

Competențe specifice:

IX.3.2 Reprezentarea algoritmilor în pseudocod

IX.4.2 Alegerea celui mai eficient algoritm de rezolvare a unei probleme

FIȘĂ DE LUCRU

Se propun următorii algoritmi pentru determinarea numărului de divizori ai numărului natural n .

1)
citește n (n număr natural)
 $nr \leftarrow 0$
pentru $i \leftarrow 1$, n execută
 dacă $n \% i = 0$ atunci
 $nr \leftarrow nr + 1$
scrie nr

2)
citește n (n număr natural)
 $nr \leftarrow 0$; $i \leftarrow 1$
cat timp $i * i \leq n$ executa
 dacă $n \% i = 0$ atunci
 $nr \leftarrow nr + 2$
 dacă $i * i = n$ atunci
 $nr \leftarrow nr - 1$
 $i \leftarrow i + 1$
scrie nr

Întrebare	Răspuns așteptat
Ce se va afișa algoritmul de la varianta 1) pentru $n=17$?	2
Ce se va afișa algoritmul de la varianta 2) pentru $n=25$?	3
Dacă pentru n se citește valoarea 1000000, câte iterații se efectuează în fiecare dintre cei doi algoritmi propuși?	1) 1000000 iterații 2) 1000 iterații
Modificați algoritmul 2) astfel încât să determine numărul divizorilor pari ai lui n .	<pre> citește n (n număr natural) $nr \leftarrow 0$; $i \leftarrow 1$ cât timp $i * i \leq n$ execută dacă $n \% i = 0$ atunci dacă $i \% 2 = 0$ && $i * i \neq n$ atunci $nr \leftarrow nr + 1$ dacă $n / i \% 2 = 0$ atunci $nr \leftarrow nr + 1$ $i \leftarrow i + 1$ scrie nr </pre>

ACTIVITĂȚI REMEDIALE CE VIZEAZĂ ASPECTELE CONSTATATE ÎN URMA APLICĂRII TESTULUI DE EVALUARE ÎNȚĂLĂ

Activitatea remedială „Lucrul cu cifrele unui număr”

Itemul 1

Competență specifică:

IX.2.3. Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.

Notă pentru profesori:

În cazul în care nu toți elevii clasei au rezolvat corect acest item, trebuie desfășurată o activitate remedială în care să fie implicați toți elevii clasei sau doar cei care nu au rezolvat corect acest item, decizia aparținând profesorului, în urma analizării situației clasei.

Fișa de lucru propusă, pleacă de la enunțul itemului 1, din testul de evaluare inițială propus. Această fișă ghidează învățarea, ajutând elevul să realizeze operații simple cu cifrele unui număr.

FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Ce operație aritmetică are simbolul ”%”?	Restul împărțirii a două numere întregi
Revenind la enunțul itemului 1, dacă $n=1234$, care va fi valoarea variabilei n după executarea atribuirii $n \leftarrow n/1000$? Ce reprezintă expresia din membrul drept al atribuirii?	Valoarea memorată în variabila n după executarea atribuirii este 1, pentru că n va memora câtul împărțirii întregi a numărului 1234 la 1000. Valoarea obținută reprezintă cifra miilor, pentru că, în conformitate cu enunțul, numărul n are exact 4 cifre.
Dacă $n=1234$, ce valoare va avea n după executarea atribuirii $n \leftarrow n/10\%10$? Ce reprezintă expresia din membrul drept al atribuirii?	$n/10\%10$ este o expresie în care operatorii au aceeași prioritate, prin urmare, operațiile se execută de la stânga la dreapta. Obținem succesiv: $1234/10\%10=123\%10$ și $123\%10=3$ Valoarea obținută reprezintă cifra zecilor.
Dacă $n=1234$, ce valoare va avea n după executarea atribuirii $n \leftarrow n\%1000/10$?	$1234\%1000/10=234/10$ $234/10=23$
Dacă $n=1234$, ce valoare va avea n după executarea atribuirii $n \leftarrow n\%1000/100$? Ce reprezintă expresia din membrul drept al atribuirii?	$1234\%1000/100=234/100$ $234/100=2$ Valoarea obținută reprezintă cifra sutelor

Activitate remedială „Alegere dificilă”

Itemul 2

Competența specifică:

IX.2.3 Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.

Notă pentru profesori:

Activitatea va avea ca scop înțelegerea modului în care lucrează operatorii logici. Va implica în principal elevii care nu au rezolvat corect acest item.

Propunem o activitate remedială, în două etape:

- *Etapa 1*: Înțelegerea relațiilor existente între date și a modului în care lucrează operatorii logici.
- *Etapa 2*: Propunerea spre rezolvare a unui alt item care verifică cunoașterea operatorilor logici.

Etapa 1

Se inițiază o activitate frontală în care profesorul propune un enunț și se solicită elevilor analizarea fiecărei variante de răspuns.

FIȘĂ DE LUCRU

1. Ioana face următoarea afirmație: *Îmi plac oamenii sportivi sau veseli.*

Întrebare	Răspuns așteptat	
În care dintre următoarele categorii poate fi o persoană pe placul Ioanei?	a. antisport și posacă	Nu o place.
	b. antisport și veselă	O place pentru că e veselă.
	c. prosport și posacă	O place pentru că e sportivă.
	d. prosport și veselă	O place pentru că are ambele calități.
	e. nici prosport și nici posacă	O place pentru că e veselă.

2. Realizați tabelele de valori pentru fiecare operator logic studiat: negație (not), conjuncție (and) și disjuncție (or).

Etapa 2

Indicați o expresie care este adevărată dacă și numai dacă expresia $x \leq 3$ or $x > 10$ este adevărată.

- $\text{not}(x < 3 \text{ and } x < 10)$
- $x \geq 3 \text{ and not}(x \geq 10)$
- $\text{not}(x < 3 \text{ or } x \leq 10)$
- $\text{not}(x > 3) \text{ or not}(x \leq 10)$

Întrebare	Răspuns așteptat	
Rescrieți expresiile folosind regulile lui De Morgan?	$\text{not}(a \text{ and } b)$	$\text{not } a \text{ or not } b$
	$\text{not}(a \text{ or } b)$	$\text{not } a \text{ and not } b$
Folosind regulile lui De Morgan identificați în ce interval se află x pentru ca expresia să fie adevărată.	$x \leq 3 \text{ or } x > 10$	$x \in (-\infty, 3] \cup (10, \infty)$
	a. $\text{not}(x < 3 \text{ and } x < 10)$	$x \geq 3 \text{ or } x \geq 10$ $x \in [3, \infty) \cup [10, \infty) \Leftrightarrow x \in [3, \infty)$
	b. $x \geq 3 \text{ and not}(x \geq 10)$	$x \geq 3 \text{ and } x < 10$ $x \in [3, \infty) \cap (-\infty, 10) \Leftrightarrow x \in [3, 10)$
	c. $\text{not}(x < 3 \text{ or } x \leq 10)$	$x \geq 3 \text{ and } x > 10$ $x \in [3, \infty) \cap (10, \infty) \Leftrightarrow x \in (10, \infty)$
	d. $\text{not}(x > 3) \text{ or not}(x \leq 10)$	$x \leq 3 \text{ or } x > 10 \Leftrightarrow$ Răspuns corect d.

Activitatea poate fi desfășurată online, folosind instrumente ce permit comunicarea sincronă, sub forma videoconferințelor (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc).

Activitate remedială „Structuri repetitive”

Itemul 3

Competențe specifice:

IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

IX.3.2 Reprezentarea algoritmilor în pseudocod.

IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.

IX.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderile de a urmări modul de execuție “linie cu linie” al unui algoritm, de a interpreta un algoritm și de a respecta principiile programării structurate.

Profesorul poate începe printr-o explicație menită să justifice alegerea răspunsului corect pentru subpunctele 3a și 3b, implicând elevii care au rezolvat corespunzător cerințele.

Activitatea propusă constă în rezolvarea unor itemi care să asigure formarea competențelor IX.3.1, IX.3.2, IX.3.3, IX.4.1 prin ghidarea învățării.

FIȘĂ DE LUCRU

Se consideră următoarea secvență de program în care toate variabilele sunt numere întregi.

```

s ← 0; i ← 3
cât timp i ≤ 7 execută
    citește x
    dacă x mod 2 ≠ 0 atunci
        s ← s + 1
    i ← i + 1
scrie s

```

Întrebare	Răspuns așteptat
Ce se va afișa dacă se citesc în ordine valorile 8 5 4 9 2?	2
Dar dacă se citesc în ordine valorile 2 4 6 8 10?	0
Ce reprezintă valoarea variabilei s , la finalul executării operațiilor?	Numărul de valori impare citite
Câte iterații se execută în structura repetitivă cât timp?	5
Rescrieți algoritmul folosind altă structură repetitivă cu număr cunoscut de pași.	<pre> s ← 0 pentru i ← 3, 7 execută citește x dacă x%2≠0 atunci s ← s + 1 scrie s </pre>
Care este rezultatul execuției următoarei secvențe dacă se citesc în ordine valorile 4 5 6 7 8?	2
<pre> s ← 0 i ← 3 repetă citește x dacă x % 2 ≠ 0 atunci s ← s + 1 i ← i + 1 până când i > 7 scrie s </pre>	
Ce ar afișa următoarele două secvențe de program, dacă s-ar citi în ordine valorile 7 5 4 9 2?	a) 0 b) 1
<p>a)</p> <pre> s ← 0 i ← 8 cât timp i ≤ 7 execută citește x dacă x % 2 ≠ 0 atunci s ← s + 1 i ← i + 1 scrie s </pre> <p>b).</p> <pre> s ← 0 i ← 8 repetă citește x dacă x % 2 ≠ 0 atunci s ← s + 1 i ← i + 1 până când i > 7 scrie s </pre>	
Modificați algoritmul de la punctul b) ca să fie echivalent cu cel de la punctul a)	Instrucțiunea "repetă-până când" se execută cel puțin o dată. Pentru echivalență se impune următoarea modificare: <pre> s ← 0 i ← 8 </pre>

	<pre> dacă i ≤ 7 atunci repetă citește x dacă x % 2 ≠ 0 atunci s ← s + 1 i ← i + 1 până când i > 7 scrie s </pre>
Formulați un enunț de problemă a cărei rezolvare să fie algoritmul dat?	La un târg expozițional care se desfășoară de miercuri până duminică, se înregistrează zilnic numărul de vizitatori. În câte zile numărul de vizitatori a fost impar?

Activitate remedială „Aplicații interdisciplinare”

Itemul 4

Competențe specifice:

IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipului datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei

IX.3.2 Reprezentarea algoritmilor în pseudocod

IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor

IX.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Notă pentru profesori:

Activitatea va avea ca scop înțelegerea modului în care trebuie abordate aplicațiile interdisciplinare. Se recomandă dezvoltarea acestei competențe în aplicații de sinteză care utilizează conținuturile „Elemente de bază și mediul de programare al limbajului Pascal sau C/C++” (Implementarea unor algoritmi elementari cu aplicabilitate practică), din programa clasei a X-a.

Propunem o activitate remedială, constând în rezolvarea unei alte probleme similare din aria curriculară a specializării. Activitatea poate avea mai multe sarcini de lucru, gradate ca nivel de dificultate, rămânând la latitudinea profesorului câte dintre acestea va aborda. O variantă ar fi ca prima cerință să se rezolve individual, iar celelalte în grupe de câte 2 elevi.

FIȘĂ DE LUCRU

1. Fie A și B două puncte din plan, identificate prin coordonatele lor. Determină distanța dintre punctele A și B și coordonatele mijlocului segmentului de dreaptă care le unește.

Întrebare	Răspuns așteptat
Câte date de intrare identificați și câte date de ieșire	Sunt patru date de intrare, reprezentând coordonatele în plan ale celor două puncte și trei date de ieșire: distanța dintre cele două puncte și coordonatele mijlocului segmentului
Folosindu-vă de teorema lui Pitagora, scrieți formula care determină distanța între două puncte A (x _a , y _a) și B (x _b , y _b)	$\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$
Realizați un algoritm pentru rezolvarea problemei	<pre> citește x_a, y_a, x_b, y_b d ← √((x_a - x_b)² + (y_a - y_b)²) x_m ← (x_a + x_b) / 2 y_m ← (y_a + y_b) / 2 scrie d, x_m, y_m </pre>

2. Răspundeți la următoarele întrebări:

Întrebare	Răspuns așteptat
Scrieți în pseudocod o instrucțiune pentru citirea coordonatelor a trei puncte din plan, A, B și C.	citește $x_a, y_a, x_b, y_b, x_c, y_c$
Scrieți o secvență de instrucțiuni pentru determinarea lungimilor segmentelor [AB], [BC] și [AC].	$a \leftarrow \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$ $b \leftarrow \sqrt{(x_b - x_c)^2 + (y_b - y_c)^2}$ $c \leftarrow \sqrt{(x_a - x_c)^2 + (y_a - y_c)^2}$
Scrieți o expresie logică care verifică dacă punctele A, B și C pot reprezenta laturile unui triunghi, cunoscând lungimile segmentelor a, b și c care le unesc.	$(a < b + c)$ și $(c < a + b)$ și $(b < a + c)$
Scrieți o expresie logică care verifică dacă punctele A, B și C sunt coliniare, cunoscând lungimile segmentelor a, b și c care le unesc.	$(a = b + c)$ sau $(c = a + b)$ sau $(b = a + c)$
Scrieți o instrucțiune care determină perimetrul triunghiului ABC, cunoscând lungimile laturilor a, b și c.	$p \leftarrow a + b + c$
Scrieți o secvență de instrucțiuni care determină aria triunghiului ABC, cunoscând lungimile laturilor a, b și c.	$p \leftarrow (a + b + c) / 2$ $\text{aria} \leftarrow \sqrt{p(p - a)(p - b)(p - c)}$

3.2. EXEMPLE DE ACTIVITĂȚI DE RECUPERARE

Activitate de învățare „Analiza problemei” – Nivel 1

Competența specifică:

IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

FIȘĂ DE LUCRU

Enunț:

Se citesc de la tastatură două numere naturale a și b ($a \leq b \leq 10000$). Realizați un algoritm care determină și afișează pe ecran numărul k de numere naturale din intervalul [a, b] cu proprietatea că sunt numere palindrom. Un număr este palindrom, dacă este egal cu oglinditul său.

Întrebare	Răspuns așteptat
Ce înțelegem prin datele de intrare ale unei probleme?	Datele de intrare sunt datele din enunțul problemei care pot fi prelucrate în cadrul algoritmului.
Care sunt datele de intrare ale acestei probleme?	a și b (reprezintă variabilele care vor memora numerele citite, conform enunțului)
Ce sunt datele de ieșire ale unei probleme?	Datele de ieșire sunt rezultatele cerute de problemă.
Care sunt datele de ieșire ale acestei probleme?	k (numărul de numere palindrom din intervalul [a, b]) reprezentând rezultatul cerut de enunț
În afara datelor de intrare și a datelor de ieșire mai pot interveni în rezolvare și alte date?	Da, datele intermediare (sau de manevră) necesare pe parcursul rezolvării.
Care sunt pașii algoritmului elementar care afișează oglinditul unui număr natural?	<i>Pas 1:</i> citim numărul x (pentru care dorim să determinăm oglinditul)

	<p><i>Pas 2:</i> inițializăm cu 0 variabila care va memora în final valoarea oglinditului</p> <p><i>Pas 3:</i> în cazul în care numărul x mai are cifre</p> <p><i>Pas 3.1:</i> identificăm ultima cifră a numărului x și o adăugăm la finalul oglinditului</p> <p><i>Pas 3.2:</i> eliminăm ultima cifră a numărului x</p> <p><i>Pas 3.3:</i> reluăm Pas 3</p> <p><i>Pas 4:</i> scriem valoarea oglinditului</p>
Care sunt pașii necesari pentru rezolvarea problemei?	<p><i>Pas 1:</i> se citesc datele de intrare a și b</p> <p><i>Pas 2:</i> se inițializează variabila k (contorul numărului de numere palindrom din intervalul $[a, b]$) cu 0.</p> <p><i>Pas 3:</i> parcurgem fiecare număr natural x, din intervalul $[a, b]$ se verifică dacă este număr palindrom și în caz afirmativ creștem cu o unitate valoarea variabilei k</p> <p><i>Pas 3.1</i> salvăm valoarea lui x într-o copie</p> <p><i>Pas 3.2</i> determinăm valoarea oglinditului lui x</p> <p><i>Pas 3.2</i> în cazul în care valoarea oglinditului este egală cu x, creștem cu o unitate valoarea lui k</p> <p><i>Pas 4:</i> se afișează valoarea variabilei k</p>

Activitate de învățare „Divizorii oglinditului” – Nivel 3

Competențe specifice:

IX.3.1 Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

IX.3.2 Reprezentarea algoritmilor în pseudocod

IX.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

IX.4.2 Alegerea celui mai eficient algoritm de rezolvare a unei probleme

Enunț:

Se citește un număr natural n . Să se determine numărul de divizori ai oglinditului lui n .

Date de intrare

Programul citește de la tastatură numărul n .

Date de ieșire

Programul afișează pe ecran numărul D , reprezentând rezultatul cerut.

Restricții și precizări

$2 \leq n \leq 1000000000$

Exemplu

Intrare	Ieșire	Explicație
63	9	Oglinditul lui 63 este 36, care are 9 divizori.

<https://www.pbinfo.ro/probleme/408/divizorii-oglintitului>

Notă pentru profesori:

Activitatea va fi structurată ca o conversație euristică, având scopul de a construi prin sarcini succesive competențele vizate. Propunem o activitate remedială, în două etape:

- *Etapa 1:* Rezolvarea itemului propus.
- *Etapa 2:* Rezolvarea unei unor cerințe suplimentare pentru același enunț.

ETAPA 1. FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Care sunt divizorii lui 12?	Divizori improprii 1 și 12 Divizori proprii 2, 3, 4 și 6
Care este oglinditul numărului 123?	321
Care sunt pașii pentru rezolvarea acestei probleme?	<i>Pas 1:</i> Se citește valoarea lui n <i>Pas 2:</i> Se determină oglinditul lui n <i>Pas 3:</i> Se determină numărul divizorilor oglinditului <i>Pas 4:</i> Se afișează numărul divizorilor determinat la pasul 3
Scrieți o secvență de instrucțiuni care permite afișarea divizorilor proprii ai numărului n ?	<pre> pentru d ← 2, [n/2] execută ┌ │ dacă n%d=0 atunci │ scrie d, "este divizor" └ </pre>
Sprrijinindu-vă pe răspunsurile anterioare scrieți un algoritm pentru determinarea divizorilor oglinditului unui număr n ?	<pre> citește n (număr natural ≥ 2) oglundit ← 0 ┌ │ cât timp n ≠ 0 executa │ oglundit ← oglundit*10 + n%10 │ n ← [n/10] └ Nrd ← 2 /*contorizăm de la început divizorii improprii ai oglinditului*/ ┌ │ pentru d ← 2, [oglundit/2] execută │ ┌ │ │ dacă oglundit%d = 0 atunci │ │ Nrd ← Nrd + 1 │ └ └ scrie Nrd </pre>

Exemplu de implementare în C++ și evaluarea pe platforma pbinfo.ro

The screenshot shows the pbinfo.ro online judge interface. At the top, there is a browser window with the URL `pbinfo.ro/probleme/408/divizorii-oglunditului/?&id_sursa=22927264#a_editor`. Below the browser, there is a code editor with the following C++ code:

```

1 #include <iostream>
2
3 using namespace std;
4 int n, og1, d=2, nr_d=2;
5 int main()
6 {
7     cin >> n;
8     og1=0;
9     while(n>0)
10        og1=og1*10+n%10;
11        n/=10;
12        for(d=2; d<=og1/2; d++)
13            if(og1%d==0)
14                nr_d++;
15        cout << nr_d;
16        return 0;
17    }

```

Below the code editor, there is a section titled "Evaluare finalizată" (Evaluation completed) showing the results of the evaluation. The table below summarizes the test results:

Test	Time	Message evaluation	Score possible	Score obtained	Example
1	0	OK	20	20	Exemplu
2	0	OK	20	20	
3	0.016	OK	20	20	
4	0	OK	20	20	
5	0	OK	20	20	

At the bottom of the table, it says "Rezultat final: 100 puncte" (Final result: 100 points).

Etapa 2

Notă pentru profesori:

Activitatea remedială poate continua cu realizarea unui algoritm echivalent optim din punct de vedere al timpului de executare

FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Cunoscând un număr n și un divizor d al său, ce putem spune despre numărul n/d ?	Și n/d este divizor al lui n . Numărul n are divizori pe d și pe n/d .
Pentru $n=12$, care sunt valorile lui d pe care le vom verifica?	$d=1$ este divizor al lui 12 și $12=12/1$ este divizorul lui 12 $d=2$ este divizor al lui 12, dar și $6=12/2$ este divizor a lui 12 $d=3$ este divizor al lui 12, dar și $4=12/3$ este divizor a lui 12
Care este valoarea maximă pe care o poate lua d , astfel încât, să determinăm corect numărul de divizori ai lui n ?	Testez dacă sunt divizori ai lui n doar valorile care respectă condiția $1 \leq d * d \leq n$
Ce se întâmplă dacă n este pătrat perfect, $d * d = n$?	În acest caz $d = n/d$ și se contorizează un singur divizor.
Sprijinindu-vă pe răspunsurile anterioare scrieți un algoritm eficient de determinare a numărului de divizori ai oglinditului lui n	<pre> citește n (n număr natural) oglindit ← 0 cat timp n ≠ 0 executa oglindit ← oglindit*10 + n%10 n ← [n/10] Nrd ← 0 d ← 1 cat timp d*d ≤ oglindit executa dacă oglindit % d=0 atunci Nrd ← Nrd + 2 dacă d*d = oglindit atunci Nrd ← Nrd - 1 d ← d + 1 scrie Nrd </pre>

Activitate de învățare „Numere prime între ele” – Nivel 3**Competențe specifice:**

IX.3.1 Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

IX.3.2 Reprezentarea algoritmilor în pseudocod

IX. 4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Algoritmi elementari**Enunț:**

Se citesc numere naturale de la tastatură până la apariția lui zero. Să se determine câte perechi de numere citite consecutiv sunt prime între ele.

Date de intrare

Programul citește de la tastatură numere naturale. Citirea se încheie la introducerea valorii 0.

Date de ieșire

Programul afișează pe ecran numărul C , reprezentând numărul de perechi citite consecutiv care respectă condiția cerută.

Restricții și precizări

- numerele citite aparțin intervalului $[0, 1\ 000\ 000\ 000)$
- valoarea zero, care încheie citirile, nu se va prelucra

Exemplu

Intrare

15 63 43 129 55 15 4
0

Ieșire

3

Explicație

Perechile care respectă condiția cerută sunt: (63, 43),
(129, 55), (15, 4).<https://www.pbinfo.ro/probleme/80/numarareperechprimeintreele>**Notă pentru profesori:**

Activitatea poate fi structurată, având scopul de a construi prin sarcini de lucru succesive competențele vizate.

FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Când spunem despre două numere naturale a și b că sunt prime între ele?	Două numere naturale a și b sunt prime între ele dacă cel mai mare divizor comun al lor este 1
Descrieți în pseudocod algoritmul lui Euclid pentru determinarea cel mai mare divizor comun a două numere naturale nenule a și b?	Algoritmul lui Euclid prin împărțiri succesive citește a, b (numere naturale nenule) r ← a%b cât timp r≠0 execută a ← b b ← r r ← a%b ■ scrie b
Care sunt pașii pentru rezolvarea problemei enunțate?	Pas 1: Se citesc primele două numere a și b Pas 2: C←0 (numărul de perechi cerut de problemă) Pas 3: În cazul în care b≠0, se efectuează: 3.1 cb←b (copiem valoarea lui b, pentru că algoritmul de determinare a celui mai mare divizor comun va modifica valoarea) 3.2 se calculează cel mai mare divizor comun al numerelor a și b 3.3 dacă b=1 atunci valoarea variabilei C crește cu o unitate 3.4 variabila a preia valoarea copiei lui b 3.5 se citește următoarea valoare din șir 3.6 se reia pasul 3 Pas 4: Se scrie valoarea variabilei C
Sprjinindu-vă pe răspunsurile anterioare, scrieți algoritmul de rezolvare a problemei	citește a, b (numere naturale nenule) C ← 0 cât timp b ≠ 0 execută cb ← b; r ← a%b cât timp r≠0 execută a ← b b ← r r ← a%b ■ dacă r=1 atunci C ← C+1 ■ a ← cb citește b ■ scrie C

Exemplu de implementare în C++ și evaluarea pe platforma pbinfo.ro

The screenshot shows the pbinfo.ro website interface. At the top, there is a search bar and navigation menus for 'Probleme', 'Soluții', and 'Resurse'. The user profile is 'Profesor' and the course is 'rodica_balasa'. The main content area displays a C++ code editor with the following code:

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int a,b,c=0,cb,r;
8     cin >> a >> b;
9     while(b != 0)
10    {
11        cb=b;
12        r = a % b;
13        while(r != 0)
14        {
15            a = b;
16            b = r;
17            r = a % b;
18        }
19        if(b == 1)
20            C++;
21        a = cb;
22        cin >> b;
23    }
24    cout << C;
25    return 0;
26 }
27

```

Below the code editor is a button labeled 'Adaugă soluția'. The results section, titled 'Evaluare finalizată', shows a table with the following data:

Test	Temp	Mesaj evaluare	Scor posibil	Scor obținut
1	0	OK	20	20
2	0	OK	20	20
3	0	OK	20	20
4	0	OK	20	20
5	0	OK	20	20

At the bottom of the results section, it states 'Rezultat final: 100 puncte'.

Activitate de învățare „Densitate” – Nivel 3

Competențe specifice:

IX.3.1 Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

IX.3.2 Reprezentarea algoritmilor în pseudocod

IX. 4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării.

Enunț:

O eprubetă plină cu apă cântărește a grame. În eprubetă se scufundă o monedă de b grame. În noua stare, eprubeta cântărește c grame. Să se determine densitatea metalului din care e confecționată moneda.

Date de intrare

Programul citește de la tastatură cele trei numere a , b și c , separate prin spații.

Date de ieșire

Programul va afișa pe ecran numărul D , reprezentând valoarea cerută.

Restricții și precizări

cele trei numere citite sunt numere reale

Exemplu**Intrare**

50 12 60.5

Ieșire

8

Explicație

Pentru $a=50$, $b=12$ și $c=60.5$ se va afișa 8 (g/cm^3).

<https://www.pbinfo.ro/probleme/3071/densitate>

FIȘĂ DE LUCRU

Întrebare	Răspuns așteptat
Cum putem determina densitatea unei substanțe când cunoaștem masa și volumul?	Formula generală este: $\text{densitate} = \text{masă} / \text{volum}$
Care este densitatea apei?	Densitatea apei = 1 kg/dm^3 , iar dacă transformăm în unitățile de măsură date de problemă, obținem densitatea apei = 1 g/cm^3
Prin adăugarea monedei în eprubetă, o parte din apă va curge. Care este volumul de apă care va curge din eprubetă?	$(a+b-c) : 1 \text{ cm}^3$
Ce reprezintă volumul apei care a curs din eprubetă?	Volumul monedei
Care este densitatea monedei?	$b / (a+b-c)$
Care este succesiunea de pași necesari pentru rezolvarea problemei?	<i>Pas 1:</i> Se citesc valorile variabilelor a, b și c <i>Pas 2:</i> volumul de apă dislocuit de monedă este $v \leftarrow a + b - c$ <i>Pas 3:</i> Densitatea monedei este $D \leftarrow b/v$ <i>Pas 4:</i> Scrie D
Sprjinindu-vă pe răspunsurile anterioare, scrieți algoritmul de rezolvare a problemei	<pre> citește a, b, c //numere reale pozitive v ← a + b - c D ← b/v scrie D </pre>

Exemplu de implementare în C++ și evaluarea pe platforma pbinfo.ro

pbInfo.ro

Probleme ▾ Soluții Resurse ▾
Profesor ▾ rodica_balasa ▾

C++ ▾

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     float a,b,c,v,D;
8     cin>>a>>b>>c;
9     v=a-(c-b);
10    D=b/v;
11    cout<<D;
12    return 0;
13 }
14

```

Adaugă soluția

Evaluare finalizată

Rezultat evaluare

Test	Time	Mesaj evaluare	Scor posibil	Scor obținut
1	0	Corect !	20	20
2	0	Corect !	20	20
3	0	Corect !	20	20
4	0	Corect !	20	20
5	0	Corect !	20	20

Rezultat final: 100 puncte

Secțiunea 4 – Adaptarea la particularitățile/categoriile de elevi în risc. Exemple de activități de învățare

Activitate de învățare „Prelucrarea cifrelor unui număr“

Competențe specifice:

IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

IX.3.2 Reprezentarea algoritmilor în pseudocod.

IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a identifica situațiile când este necesară folosirea structurii alternative și de a o folosi corect pe aceasta.

Activitatea propusă constă în alegerea primei lecții din setul de lecții AEL, „Algoritmi elementari pentru prelucrarea numerelor” și rezolvarea testelor din secțiunea de evaluare.

<https://eduonline.roedu.net/repository/pl-tic-6-5-2%20lectii/>

Algoritmi elementari pentru prelucrarea numerelor

- Captarea atenției
- 1. Operatorii DIV și MOD. Determinarea cifrelor unui număr
- 2. Algoritm de numărare a cifrelor unui număr natural
- 3. Recompunerea unui număr pornind de la cifrele lui
- 4. Evaluare
- 5. Analiza unui algoritm dat (1)
- 6. Analiza unui algoritm dat (2)
- 7. Determinarea unui număr aflat într-un interval dat
- 8. Evaluarea metodei prezentată anterior

Algoritmi elementari pentru prelucrarea numerelor

Algoritm de numărare a cifrelor unui număr natural

Numărul ales este: 45192

Simulare

număr 4 5 1

contor 2

algoritm în pseudocod

```

citește număr
contor ← 0
cât timp număr ≠ 0 execută
| număr ← număr DIV 10
| contor ← contor + 1
scrie contor

```

Activitatea remedială „Structura repetitivă cu test inițial”

Competențe specifice:

IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

IX.3.2 Reprezentarea algoritmilor în pseudocod.

IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.

Notă pentru profesori:

În cazul în care se constată la unii elevi carențe în dezvoltarea competențelor anterioare, se recomandă dezvoltarea acestora în aplicații legate de „Elemente de bază și mediul de programare al limbajului Pascal sau C/C++” (Structuri de control- Structuri repetitive), din programa clasei a X-a.

Activitatea poate fi organizată frontal, cu implicarea cu precădere a elevilor care necesită suport remedial. Activitatea își propune să dezvolte la elevi deprinderea de a urmări modul de execuție “linie cu linie” al unui algoritm, cu scopul de a înțelege funcționarea structurii repetitive.

Item 1

Competența specifică:

IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

FIȘĂ DE LUCRU

De câte ori se repetă instrucțiunile din structura repetitivă `cat_timp` și care va fi valoarea variabilei `i` la sfârșitul următoarei secvențe de algoritm?

```
i ← 1
s ← 0 (numere întregi)
cât timp i ≤ 5 execută
  s ← s + i
  i ← i + 1
```

- a. se obține o buclă infinită
b. de 5 ori și $i=6$
c. de 6 ori și $i=6$
d. de 5 ori și $i=5$

Pentru rezolvarea sarcinii de lucru, realizați un tabel în care să se păstreze, pe fiecare coloană, valorile variabilelor modificate în cadrul fiecărei iterații

<code>i</code>	1					
<code>s</code>	0					
<code>i ≤ 5</code>	true	true	true	true	true	false
<code>s ← s + i</code>	1	3	6	10	15	
<code>i ← i + 1</code>	2	3	4	5	6	

Răspuns corect: de 5 ori și $i=6$

Item 2

Competența specifică:

IX.3.2 Reprezentarea algoritmilor în pseudocod.

FIȘĂ DE LUCRU

Identificați greșeala din următoarea secvență de instrucțiuni care ar trebui să afișeze primele 10 numere naturale?

```
i ← 0
cât timp i ≤ 9 execută
  scrie i
```

Pas 1: se simulează execuția algoritmului linie cu linie, realizând un tabel în care se păstrează, pe fiecare coloană, valorile variabilelor modificate în cadrul fiecărei iterații

<code>i</code>	0				
<code>i ≤ 9</code>	true	true	true	true	... true
<code>scrie i</code>	0	0	0	0	... 0

Pas 2: se deduce că structura are un număr infinit de iterații deoarece variabila `i` nu își modifică valoarea. Inserăm în structura repetitivă atribuirea `i ← i+1`, apoi se poate verifica rezultatul.

<code>i</code>	0				
<code>i ≤ 9</code>	true	true	true	true	... false
<code>scrie i</code>	0	1	2	3	... 9
<code>i ← i+1</code>	1	2	3	4	... 10

Răspuns corect: `i ← i+1`

Item 3**Competența specifică:**

IX.3.2 Reprezentarea algoritmilor în pseudocod.

FIȘĂ DE LUCRUDacă n este un număr natural $0 < n < 10$, ce realizează următoarea secvență?

```

p ← 1
cât timp n > 0 execută
    p ← p * n
    n ← n - 1

```

- suma numerelor mai mici sau egale ca n
- produsul cifrelor numărului natural n
- produsul numerelor mai mici sau egale ca n

Presupunem că $n=4$. Realizați un tabel în care se păstrează, pe fiecare coloană, valorile variabilelor modificate în cadrul fiecărei iterații a structurii repetitive `cat_timp`.

n	4				
p	1				
n > 0	true	true	true	true	false
p ← p * n	4	4 * 3	4 * 3 * 2	4 * 3 * 2 * 1	
n ← n - 1	3	2	1	0	

Răspuns corect: produsul numerelor mai mici sau egale ca n .**Item 4****Competențe specifice:**

IX.3.2. Reprezentarea algoritmilor în pseudocod.

FIȘĂ DE LUCRUCare trebuie să fie valoarea inițială a unei variabile întregi i pentru ca secvența următoare să afișeze 11111?

```

i ← ?
cât timp i ≠ 1 execută
    i ← i + 1
    i ← i - 2
    scrie 11

```

Întrebare	Răspuns așteptat
Câte iterații trebuie să efectueze structura <code>cat_timp</code> pentru a se afișa 11111?	3
Cum se modifică valoarea lui i în buclă?	Scade cu 1.
Care trebuie să fie valoarea inițială a lui i pentru a se afișa 11111?	Răspuns corect: $i=4$

Item 5**Competențe specifice:****Item 5a.** IX.3.1 Analizarea enunțului unei probleme: identificarea datelor de intrare și a datelor de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.**Item 5b.** IX.3.2 Reprezentarea algoritmilor în pseudocod.

FIȘĂ DE LUCRU

Pentru următoarea secvență de algoritm:

```

citește x
cât timp x>0 execută
  dacă x%2 = 0 atunci
    scrie x%10
  x←[x/10]

```

a. Ce se va afișa dacă pentru variabila x se va citi valoarea 27843?

Simulați execuția algoritmului linie cu linie, realizând un tabel în care se păstrează, pe fiecare coloană, valorile variabilelor modificate în cadrul fiecărei iterații

x	27843					
$x > 0$	true	true	true	true	true	false
$x \% 2 = 0$	F	T	T	F	T	
scrie $x \% 10$	-	4	8	-	2	
$x \leftarrow [x/10]$	2784	278	27	2	0	

Răspuns corect: Algoritmul afișează cifrele pare ale lui x , în ordinea în care acestea apar în număr, de la dreapta la stânga.

b. Dați exemplu o valoare de intrare pentru care algoritmul nu afișează nimic.

Întrebare	Răspuns așteptat
Ce observație puteți să faceți în legătură cu rezultatul afișării?	Se afișează doar cifre pare.
Cum ar trebuia să fie x pentru a nu se afișa nimic?	x ar trebui să aibă doar cifre impare.
Dați exemplu o valoare de intrare pentru care algoritmul nu afișează nimic	Răspuns corect: $x=3517$ sau orice număr care conține doar cifre impare

Item 6**Competențe specifice:**

Item 6a. IX.3.2 Reprezentarea algoritmilor în pseudocod.

Item 6b. IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.

FIȘĂ DE LUCRU

Considerăm următoarea secvență, în care toate variabilele care intervin sunt întregi.

```

s ← 0
i ← 1
cât timp i ≤ 7 execută
  citește x
  s ← s + x
  i ← i + 1

```

Întrebare	Răspuns așteptat
Câte numere se vor citi?	7
Ce vor memora, la finalul algoritmului, variabilele s , respectiv i ?	s calculează suma valorilor citite i contorizează numărul valori citite
Ce operație realizează algoritmul prezentat?	Citirea și însumarea valorilor dintr-un șir format din 7 numere.
Scrieți un algoritm echivalent, folosind altă structură repetitivă?	<pre> s ← 0 pentru i ← 1, 7 execută citește x s ← s + x </pre>

Activitate de învățare „Structura alternativă (exemple și exerciții)”

Competențe specifice:

IX.3.2 Reprezentarea algoritmilor în pseudocod.

IX.3.3 Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a identifica situațiile când este necesară folosirea structurii alternative și de a o folosi corect pe aceasta.

Activitatea propusă constă în parcurgerea lecției AEL “Structura alternativă (exemple și exerciții)”, din setul de lecții „Gândirea algoritmică” și rezolvarea testelor din secțiunea de evaluare.

<http://info.tm.edu.ro:8080/~dorin/public/TIC/PL-TIC-6-2-5%20lectii/>

Gândirea algoritmică

- Captarea atenției
- 1. Noțiunea de algoritm
- 2. Exemple de algoritmi
- 3. Evaluare (1)
- 4. Descrierea algoritmilor
- 5. Exerciții
- 6. Evaluare (2)
- 7. Structuri (Partea I)
- 8. Structura liniară (exemple și exerciții)
- 9. Evaluare (3)
- 10. Structuri (Partea a II-a)
- 11. **Structura alternativă (exemple și exerciții)**
- 12. Evaluare (4)
- 13. Structuri (Partea a III-a)
- 14. Structura repetitivă (exemple și exerciții)
- 15. Evaluare (5)

G

A

E

Gândirea algoritmică

AeL
eCONTENT
Built to teach intelligently

Structura alternativă (exemple și exerciții)

?

Cerință

Fie trei localități: A, B, C și:
a - distanța de la A la B;
b - distanța de la B la C;
c - distanța de la C la A.

Vizitează toate cele trei orașe, pornind din oricare dintre orașele A sau B, ajungând în final, în oricare dintre orașele B sau C, alegând un traseu de lungime minimă.

Distanțe

Ruta:
Distanța parcursă: 0

Clasa a XI-a (matematică-informatică)

Secțiunea 1 – Repere pentru estimarea nivelului achizițiilor învățării la finele anului școlar 2019-2020 în vederea realizării planificării calendaristice pentru noul an școlar

În vederea elaborării planificării calendaristice se recomandă următorul demers:

- se analizează programele școlare în vigoare, clasele a X-a și a XI-a, disciplina Informatică
 - o programa clasei a X-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_clasa%20a%20X-a.pdf
 - o programa clasei a XI-a
http://programe.ise.ro/Portals/1/Curriculum/Progr_Lic/TH/Informatica_teoretic_vocational_clasa%20a%20XI-a.pdf
- din programa clasei a X-a se extrag acele competențe specifice a căror structurare ar fi trebuit realizată preponderent în semestrul al II-lea (în perioada în care învățarea s-a realizat la distanță) și conținuturile aferente;
- se analizează competențele specifice din programa clasei a XI-a și se stabilesc legături între acestea și competențele nestructurate/ parțial structurate în anul școlar 2019-2020;
- se analizează conținuturile neabordate/parțial abordate din clasa a X-a și se stabilește modul de fuzionare cu elementele de conținut aferente clasei a XI-a, astfel încât logica internă a domeniului să fie păstrată în condițiile structurării competențelor specifice ale clasei a XI-a.

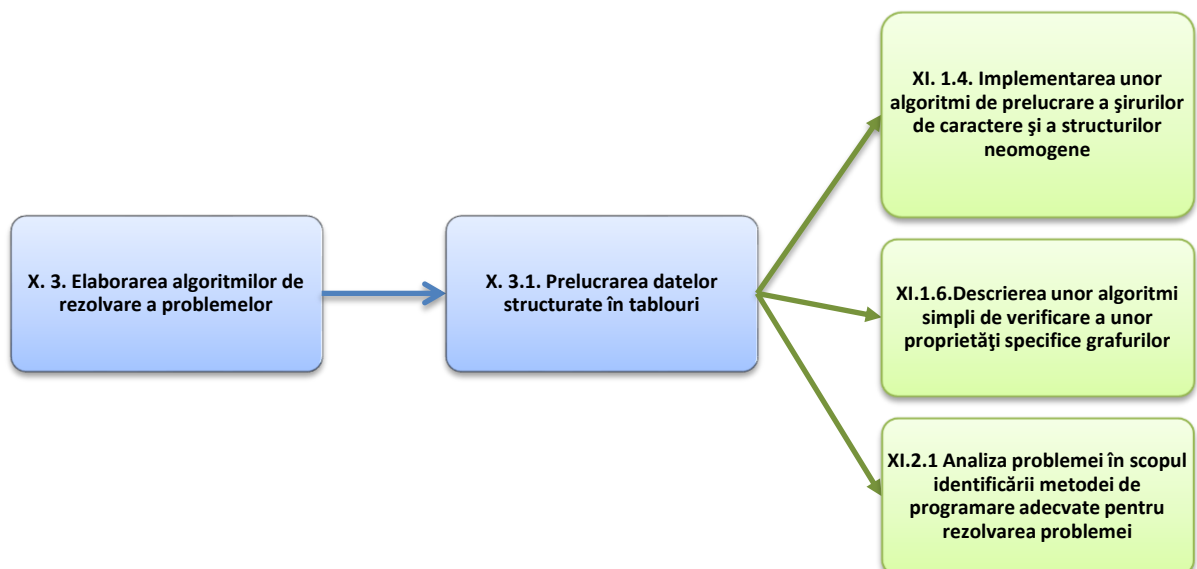
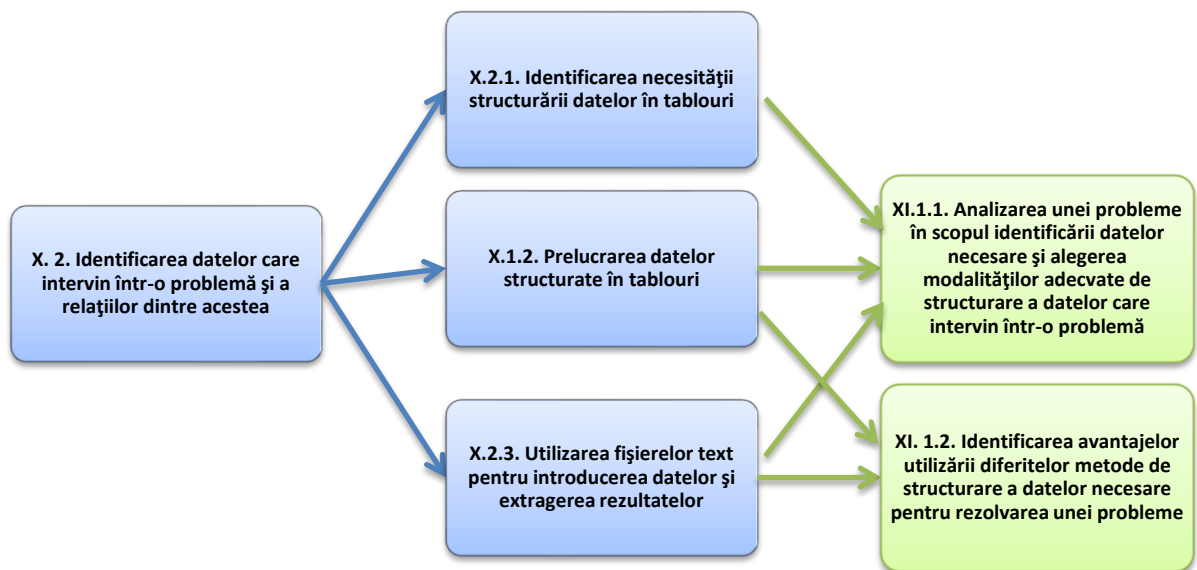
La analiza programei școlare și a planificării calendaristice a anului școlar 2019-2020, identificăm unitățile de învățare care au fost desfășurate prin activități derulate exclusiv online. Competențele specifice corespunzătoare sunt X.2.1, X.2.2, X.3.1, X.4.1, X.4.2, X.5.1, X.5.2. În anexa 2 sunt prezentate aceste competențe și sunt reținute conținuturile, care sunt corelate cu parcursul școlar din perioada vizată de această analiză.

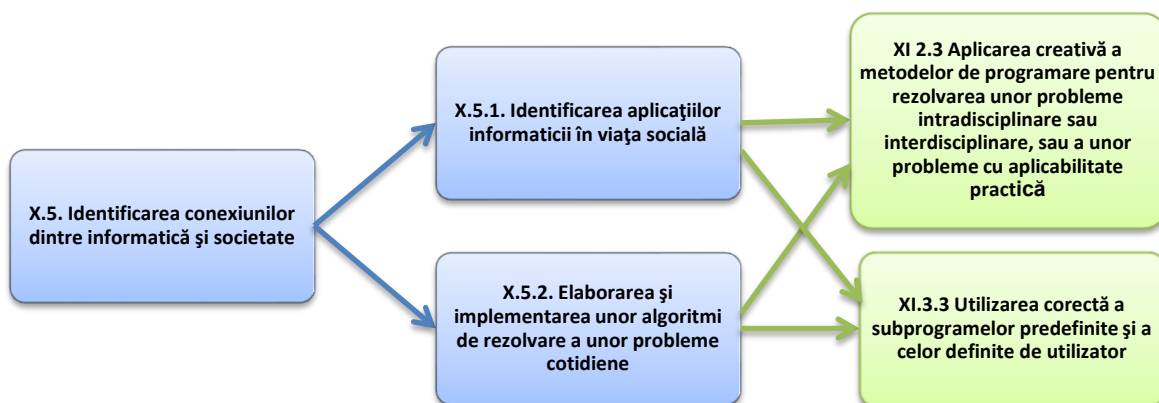
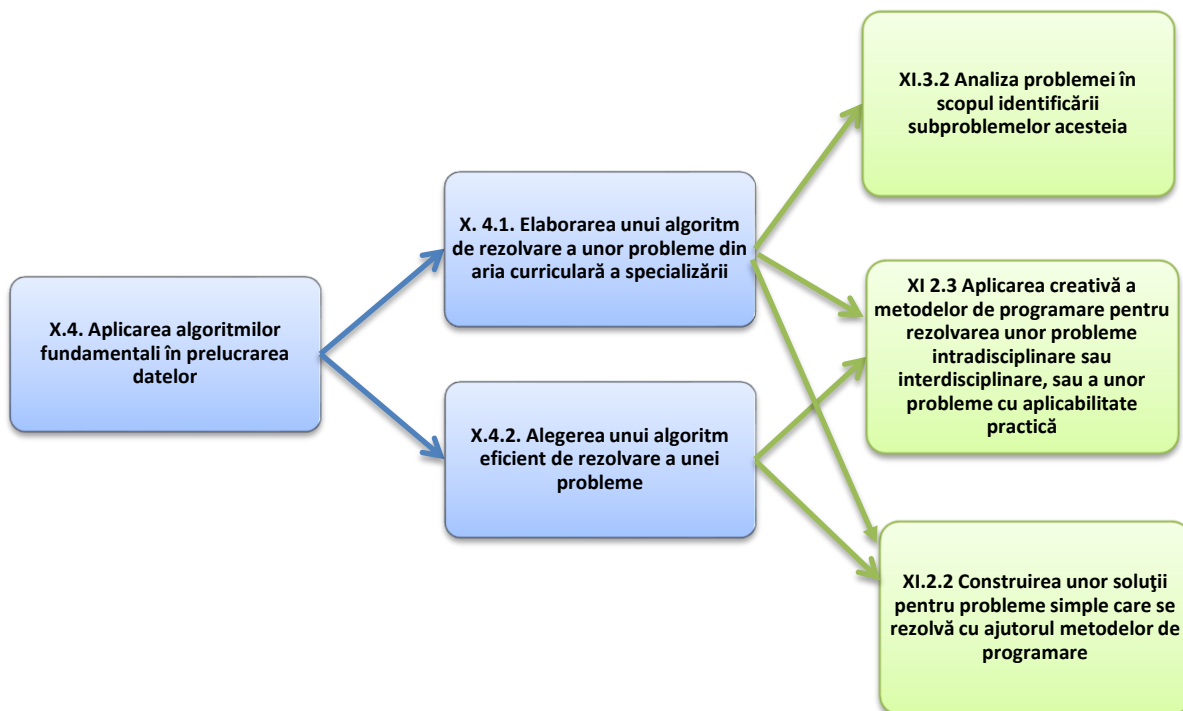
Competențele X.2.1, X.2.2 și X.3.1 de la clasa a X-a, dezvoltate cu ajutorul conținuturilor **Tipuri structurate de date** (Tipul tablou și algoritmi fundamentali de prelucrare a datelor structurate în tablouri) pot fi structurate cu ajutorul conținuturilor **Șiruri de caractere**, **Structuri de date neomogene (struct/record)** și **Liste** din clasa a XI-a. De exemplu, conținuturile legate de tablouri unidimensionale și bidimensionale pot fi reluate și fixate în cadrul noilor tipuri de date structurate introduse în clasa a XI-a. De asemenea, în cadrul conținuturilor **Aplicații folosind subprograme** pot fi folosite exemple care impun lucrul cu tablouri.

Competența X.2.3 va fi dezvoltată pe parcursul întregului an școlar, solicitând la problemele propuse ca datele de intrare să fie preluate din fișiere, respectiv datele de ieșire să fie memorate în fișiere.

Competențele X.4.1 și X.4.2 se vor structura, în anul școlar 2020-2021, în procesul de dezvoltare a competențelor XI.2.2, XI.2.3 și care dezvoltă elevilor capacitatea de proiectare, analiză și optimizare a algoritmilor de rezolvare a unor probleme din aria curriculară a specializării, și respectiv XI.3.2 în care elevii învață să analizeze o problemă, să o descompună în subprobleme pe care să le rezolve cu ajutorul subprogramelor.

Competențele X.5.1 și X.5.2 pot fi dezvoltate în anul școlar 2020-2021, cu ajutorul competențelor XI 2.3 și XI 3.3. Conținuturile **Șiruri de caractere**, **Structuri de date neomogene (struct/record)** - Rezolvarea unor probleme cu caracter practic, **Liste**, **Grafuri orientate și neorientate** respectiv **Subprograme** vor fi suport pentru aceasta. Pentru atingerea tuturor competențelor, profesorul va încerca prioritizarea conținuturilor, dirijând învățarea în funcție de gradul de progres al elevilor. În continuare este prezentată, ca model, o modalitate în care se poate continua dezvoltarea competențelor vizate de aceste repere metodologice, de-a lungul clasei a XI-a, conform programei școlare de Informatică filiera teoretică, profil real, specializarea Matematică-informatică.





Secțiunea 2 – Evaluarea gradului de achiziție a competențelor din anul anterior

EXEMPLU DE TEST DE EVALUARE ÎNȚĂLĂ

Matricea de specificații:

	Nivelul 1 (Cunoaștere)	Nivelul 2 (Aplicare)	Nivelul 3 (Raționament)
X.1.1 Implementarea algoritmilor reprezentați în pseudocod în limbaj de programare	(4c)		(4d)
X.1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)	(3)	(5b)	
X.2.1 Identificarea necesității structurării datelor în tablouri.	(1)	(5a)	
X.3.1 Prelucrarea datelor structurate	(2), (4a)	(4b)	
X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării		(6a)	(5c)
X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme		(6c)	(6b)

Itemul 1

Competența specifică:

X.2.1 Identificarea necesității structurării datelor în tablouri.

Se citesc de la tastatură 10 de numere naturale. Care dintre următoarele cerințe presupun realizarea unui algoritm care utilizează date structurate – tablouri unidimensionale?

- Determinați suma numerelor mai mari decât primul număr citit.
- Determinați suma dintre cel mai mic și cel mai mare număr citit
- Determinați suma numerelor mai mari decât ultimul număr citit.
- Determinați suma numerelor pare citite.

Răspuns corect: c

Itemul 2

Competența specifică:

X.3.1 Prelucrarea datelor structurate în tablouri

Considerăm un tablou bidimensional cu N linii și M coloane. Câte elemente sunt situate pe marginea tabloului (prima și ultima linie, prima și ultima coloană)?

- $2 * M + 2 * N$
- $M * N - 2 * N - 2 * M$
- $2 * N + 2 * M - 4$
- $2 * N + 2 * M - 2$

Răspuns corect: c

Itemul 3

Competența specifică:

X.1.2 Prelucrarea datelor structurate în tablouri

Fie declarația: *Varianta Pascal*

```
var a : array[0..2, 0..2] of byte; i, j : byte;
```

Varianta C/C++

```
int i, j, a[3][3];
```

Specificați care vor fi elementele tabloului după execuția secvenței următoare de instrucțiuni:

<i>Varianta Pascal</i>		<i>Varianta C/C++</i>	
for i:=0 to 2 do for j:=0 to 2 do a[i,j]:=i + j;		for (i=0; i<3; i++) for (j=0; j<3; j++) a[i][j]= i + j;	
a. 1 1 1	b. 0 1 2	c. 0 1 2	d. 1 2 3
2 2 2	1 2 3	0 1 2	2 3 4
3 3 3	2 3 4	0 1 2	3 4 5

Răspuns corect: b

Itemul 4

Competențe specifice:

Item 4.a X.3.1 Prelucrarea datelor structurate

Item 4.b X.3.1 Prelucrarea datelor structurate

Item 4.c X.1.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare

Item 4.d X.1.1 Transcrierea algoritmilor din pseudocod într-un limbaj de programare

Se consideră următorul program pseudocod alăturat:

- Ce se va afișa pentru $N=6$ și tabloul unidimensional $A=\{2, 0, 1, 4, 6, 3\}$?
- Dați un exemplu de set de date de intrare pentru care nu se va efectua nici o interschimbare.
- Transcrieți în limbajul de programare studiat, algoritmul alăturat. Presupunem că tabloul A reține notele elevilor unei clase, la testul de informatică.
- Modificați algoritmul pentru a afișa pe ecran, șirul notelor ordonate descrescător.

```

citeste N;
[ pentru i←1, N executa
  citeste A[i];
]
[ pentru i←1, N-1 executa
  [ dacă A[i]>A[i+1] atunci
    A[i] ↔ A[i+1];
  ]
]
[ pentru i←1, N executa
  scrie A[i];
]

```

Răspuns corect 4a: 0 1 2 4 3 6

Răspuns corect 4b: răspunsul trebuie să fie un șir de valori ordonate crescător

Răspuns corect 4c:

O implementare posibilă ar fi:

```

#include <iostream>
using namespace std;
int A[100], N;
int main(){
  cin>> N;
  for (int i=1; i<=N; i++) cin>>A[i];
  for (int i=1; i<N; i++)
    if (A[i]>A[i+1]) {int c = A[i]; A[i]=A[i+1]; A[i+1]=c;}
  for (int i=1; i<=N; i++) cout<<A[i]<<" ";
  return 0;
}

```

Răspuns corect 4d:

Elevul poate alege orice metodă de sortare învățată: sortarea cu bule, prin inserție, prin selecție sau numărare. O variantă de algoritm ar putea fi:

```

citeste N;
[ pentru i ← 1, N executa
  citeste A[i];
]

```

```

pentru i ← 1, N-1 executa
  p ← i
  pentru j ← i+1, N executa
    dacă A[i]>A[p] atunci
      p ← j
  A[i] ↔ A[p]
pentru i ← 1, N executa
  scrie A[i];

```

Itemul 5

Competențe specifice:

Item 5.a X.2.1 Identificarea necesității structurării datelor în tablouri.

Item 5.b X.1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

Item 5.c X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Se consideră un tablou bidimensional cu N linii și M coloane, având elemente numere întregi. De pe prima linie a fișierului `test.in` se citește perechea de numere naturale N și M , de cel mult 2 cifre, iar de pe următoarele N linii, câte M numere naturale reprezentând elementele matricei.

- Scrieți o declarație de variabile necesare pentru memorarea datelor de intrare.
- Scrieți o secvență de program care permite citirea datelor de intrare.
- Scrieți o secvență de program care determină numărul de linii care au elementele dispuse în ordine strict crescătoare.

Răspuns corect 5a: `int A[100][100], N, M;`

Răspuns corect 5b:

```

ifstream fin("test.in");
fin>>N>>M;
for (int i=1; i<=N; i++)
    for (int j=1; j<=M; j++) fin>>A[i][j];

```

Răspuns corect 5c:

```

int nr = 0;
for (int i=1; i<=N; i++) {
    bool ok = true;
    for (int j=1; j<M; j++)
        if (A[i][j]>=A[i][j+1]) ok = false;
    if (ok) nr++;
}
cout<<nr<<" ";

```

Itemul 6

Competențe specifice:

Item 6.a X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Item 6.b X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Item 6.c X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Se consideră două șiruri ordonate crescător, primul format din N numere naturale, respectiv M numere naturale, toate având cel mult 9 cifre, memorate în doi vectori. Se dorește afișarea, în ordine **strict** crescătoare, a valorilor pare din cele două șiruri.

- Descrieți în limbaj natural un algoritm care folosește algoritmul de sortare pentru rezolvarea cerinței
- Scrieți o secvență de program folosind un algoritm eficient din punct de vedere al timpului de executare.
- Justificați eficiența algoritmului eficient propus.

Răspuns corect 6a:

Valorile pare din cele două șiruri pot fi memorate într-un vector, care va fi sortat prin una dintre metodele învățate (selecție, inserție, metoda bulelor). După această operație, elementele de valori egale se află, plasate în vector, pe poziții alăturate. Începând cu primul element, vor fi afișate toate elementele care sunt diferite de elementul alăturat situat pe poziția precedentă ($A[i-1] \neq A[i]$).

Răspuns corect 6b:

O variantă de implementare poate fi:

```
int u = 0, i, j, t;
i=j=1;
while (i<=N && j<=M)
    if (A[i]%2) i++;
    else if (B[j]%2) j++;
    else if (A[i]<B[j]){
        if (A[i]>u) cout<<A[i]<<" ", u=A[i];
        i++;}
    else{
        if (B[j]>u) cout<<B[j]<<" ",u=B[j];
        j++;}
for (t=i; t<=N; t++)
    if (A[t]>u&& A[t]%2==0) cout<<A[t]<<" ", u=A[t];
for (t=j; t<=M; t++)
    if (B[t]>u&& B[t]%2==0) cout<<B[t]<<" ", u=B[t];
```

Răspuns corect 6c:

Algoritmul propus are la bază principiul algoritmului de interclasare. Complexitatea acestuia este liniară $O(N+M)$, pe când algoritmul propus la punctul a. are o complexitate pătratică $O((N+M)^2)$, generată de algoritmul de sortare.

Timp de lucru propus: **90** minute

Notă pentru profesori:

Se va urmări ca în cadrul testului de evaluare inițială să existe minimum doi itemi pentru fiecare competență specifică evaluată.

Secțiunea 3 – Recomandări pentru construirea noilor achiziții. Exemple de activități de învățare

3.1. EXEMPLE DE ACTIVITĂȚI DE ÎNVĂȚARE REMEDIALE

Activitate remedială „Urna cu Bile”

Itemul 1

Competența specifică:

X.2.1 Identificarea necesității structurării datelor în tablouri.

Notă pentru profesori:

În situația în care activitatea se desfășoară în clasă, atunci profesorul ar putea începe printr-o explicație menită să justifice alegerea răspunsului c. Profesorul poate reprezenta algoritmi corespunzători fiecărei variante de răspuns, implicând elevii care au rezolvat corect. De asemenea, poate iniția o activitate frontală, structurată sub forma unei conversații euristice, în care acesta propune un enunț scurt, pentru ca mai apoi să solicite elevilor modificarea cerinței astfel încât să fie necesară folosirea datelor structurate de tip tablou. Elevii trebuie încurajați să propună cât mai multe variante de cerințe a căror rezolvare necesită utilizarea datelor structurate de tip tablou unidimensional, eventual să prezinte, pe caiete, câte un algoritm pentru una dintre ele.

Activitatea poate fi desfășurată online, folosind instrumente ce permit comunicarea sincronă, sub forma videoconferințelor (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc).

FIȘĂ DE LUCRU

Enunț	Cerința inițială	Variantă posibilă
Într-o urnă sunt introduse, succesiv, 100 de bile, fiecare dintre ele având inscripționat un număr natural.	Câte bile au avut același număr cu cel inscripționat pe prima bilă?	Câte bile au avut același număr cu cel inscripționat pe ultima bilă?

Activitate remedială „Organizarea datelor într-un tablou bidimensional”

Itemul 2

Competența specifică:

X.3.1 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea va viza înțelegerea modului de organizare a datelor într-un tablou bidimensional. Profesorul poate iniția o conversație euristică construită ca o succesiune de întrebări, ale căror răspunsuri vor ajuta elevul la înțelegerea reprezentării conceptuale a tablourilor bidimensionale.

Activitatea poate fi desfășurată online, folosind platforme care permit realizarea testelor.

FIȘĂ DE LUCRU

Pornind de la enunțul itemului 2, considerăm un tablou bidimensional A cu 10 linii și 15 coloane

Întrebare	Răspuns așteptat
Câte elemente sunt memorate în tabloul A?	$10 \times 15 = 150$
Câte elemente sunt situate în tabloul A pe prima linie?	15
Câte elemente sunt situate în tabloul A pe ultima coloană?	10
Câte elemente rămân în tabloul A dacă se elimină ultima linie?	$9 \times 15 = 135$
Câte elemente ale tabloului A au un număr minim de elemente vecine (alăturate pe linie sau pe coloană)?	4
Câte elemente sunt situate pe marginea tabloului A?	$10 + 10 + 15 + 15 - 4$

Activitate remedială „Completarea elementelor într-un tablou bidimensional”

Itemul 3

Competențe specifice:

X.3.1 Prelucrarea datelor structurate

X.1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

Notă pentru profesori:

Activitatea va urmări dezvoltarea competenței de prelucrare a datelor structurate, respectiv parcurgerea și referirea elementelor unui tablou bidimensional, dar și utilizarea unui mediu de programare, la nivel de cunoaștere. Ca sarcină de lucru, elevii vor fi sprijiniți să rezolve un item de asociere simplă. Activitatea va putea fi organizată în clasă, ca o activitate frontală, în care sunt implicați în primul rând elevii care nu au rezolvat corect itemul 3.

Întreaga activitate poate fi însă desfășurată și online, folosind instrumente de comunicare sincrone cu elevii (Google Meet, Zoom, Facebook Messenger, Skype, Webex, etc), profesorul având posibilitatea prezentării itemului și rezolvării acestuia.

FIȘĂ DE LUCRU

1. Desenați un tabel care corespunde unei matrice pătratice cu 4 linii și 4 coloane. Hașurați elementele care sunt plasate pe diagonala principală, respectiv sunt plasate pe diagonala secundară.
2. Realizați asocierile corecte între secvențele de instrucțiuni (coloana din stânga) și valorile elementelor tabloului A (coloana din dreapta), considerând următoarea declarație de variabile:

```
int A[4][4], i, j;
```

1)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        A[i][j]=i;
```

2)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        A[i][j]=j;
```

3)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        if (i==j) A[i][j]=2*i;
```

4)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        if (i==j) A[i][j]=i+j;
```

5)

```
for (i=0; i<4; i++)
    for (j=0; j<4; j++)
        if (i==3 || j==3) A[i][j]=abs(i-j);
```

6)

```
for (i=0; i<4; i++) {
    A[i][3]=abs(i-j);
    A[3][i]=abs(i-j);
}
```

7)

```
for (i=0; i<4; i++) A[i][i]= 2*i;
```

8)

```
for (i=0; i<4; i++) A[3][i]=A[i][3]=3-i;
```

a)

0	0	0	3
0	0	0	2
0	0	0	1
3	2	1	0

b)

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

c)

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

d)

0	0	0	0
0	2	0	0
0	0	4	0
0	0	0	6

Activitate remedială „Regulă de atribuire“

Itemii 4a și 4b

Competența specifică:

X.3.1 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a urmări modul de execuție “linie cu linie” al unui algoritm (program). Elevul va fi sprijinit să învețe parcurgerea și referirea elementelor unui tablou unidimensional.

Activitatea propusă constă în rezolvarea unei fișe de lucru ce conține itemi de completare. Activitatea va fi individuală și poate fi desfășurată în clasă, dar și online.

FIȘĂ DE LUCRU

Considerăm declarația:

```
int A[100], i;
```

1. Scrieți o expresie logică care verifică dacă indicele elementului $A[i]$ este par.
2. Completați punctele de suspensie din următoarea afirmație: Elementele situate pe poziții alăturate cu $A[i]$ sunt ...
3. Completați punctele de suspensie din secvențele următoare astfel încât elementele tabloului A să fie egale cu:

3	3	5	5	7	7	9	9	11	...	101
A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	...	A[99]

a.

```

pentru i ← 0, .. executa
    [daca i%2=0 A[i] ← ...
     altfel A[i] ← ...
    ]

```

b.

```

pentru i ← 0, ..., 2 executa
    A[i] ← ...
    A[i+1] ← ...

```

c.

```

pentru i ← 0, ... executa
    A[i] ← ...

```

Activitate remedială „Joc online “

Itemul 4c

Competențe specifice:

X.1.1 Implementarea algoritmilor reprezentați în pseudocod în limbaj de programare

Notă pentru profesori:

Activitatea remedială își propune să dezvolte la elevi deprinderea de a transcrie corect un algoritm în limbaj de programare, să cunoască structura, elementele de bază, instrucțiunile și sintaxa limbajului studiat.

Activitatea propusă constă în alegerea unei aplicații educaționale și parcurgerea individuală a acesteia. De exemplu, pe platforma educațională .campion se poate alege unul dintre jocurile propuse (Labirint, Rebus, Aritmogrif, sau Pescuitorul de perle), în cadrul aplicației “Elemente de bază ale limbajului C/C++” http://campion.edu.ro/arhiva/www/arhiva_2009/seds/10/index.htm

Elemente de bază ale limbajului C/C++

Index

Lecția 1

- Setul de caractere
- Identificatori
- Cuvinte rezervate
- Constantele întregi
- Constantele reale
- Constantele caracter
- Constantele șir de caractere

Lecția 2

- Codificarea numerelor naturale
- Codificarea numerelor întregi
- Codificarea caracterelor
- Codificarea numerelor reale
- Variabile

Lecția 3

- Expresii
- Evaluarea expresiilor
- Tipul expresiilor
- Operatori aritmetici
- Operatori de incrementare/decrementare

Lecția 4

- Operatori relaționali
- Operatori de egalitate
- Operatori logici globali
- Operatori logici pe biți

Lecția 5

- Operatori de atribuire
- Operatori condiționali
- Operatorul de conversie explicită de tip
- Operatorul de determinare a dimensiunii
- Operatorul virgulă

Lecția 6

- Joc didactic Labirint
- Joc didactic Rebus
- Joc didactic Aritmogrif
- Joc didactic Pescuitorul de perle

Lecția 7

Fixarea cunoștințelor
Evaluare finală

Periețanu Alexandru
Răducanu Dragoș

prof. Emanuela Cerchez
prof. Marinel Șerban

O aplicație gratuită, utilă pentru dezvoltarea capacității de a transcrie și de a realiza programe în C++ este aplicația *Learn C++* disponibilă pe Android și pe iOS.

Activitate remedială „Ordonare prin selecție”

Itemul 4d

Competențe specifice:

X.1.1 Implementarea algoritmilor reprezentați în pseudocod în limbaj de programare

X.2.1 Identificarea necesității structurării datelor în tablouri.

X.3.1 Prelucrarea datelor structurate

Notă pentru profesori:

Activitatea remedială își propune să ajute elevii să își dezvolte capacitatea de prelucrare a datelor structurate, să înțeleagă și să utilizeze corect algoritmi de sortare. Ea poate fi realizată atât frontal, cât și online. Dacă activitatea se va desfășura online, se vor putea folosi instrumentele sincrone de comunicare cu elevii.

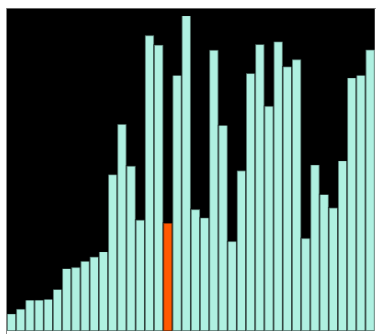
FIȘĂ DE LUCRU

Urmăriți exemplul și animația care descrie pașii algoritmului de sortare prin selecție disponibile la adresa <https://www.pbinfo.ro/articole/5605/sortarea-prin-selectie#intlink-3>

Algoritmul descris mai sus se mai numește **sortare prin selecție generală**, sau **implicită**. O altă variantă este următoarea, în care pentru fiecare secvență $i \dots n-1$ se determină explicit minimul și se interschimbă cu $x[i]$.

```
int n, X[100];
//citire X[] cu n elemente
for(int i = 0 ; i < n - 1 ; i ++ )
{
    int p = i;
    for(int j = i + 1 ; j < n ; j ++ )
        if(X[j] < X[p])
            p = j;
    int aux = X[i];
    X[i] = X[p];
    X[p] = aux;
}
```

Animație



[Vezi animația aici!](#)

Resurse online

[Selection Sort pe Wikipedia](#)

[Animație pas cu pas](#)

[Animație, valori configurabile](#)

După vizionarea animației, rezolvați următoarele sarcini de lucru:

1. Considerăm un tablou unidimensional $x = \{3, 1, 7, 4, 9\}$. Scrieți pe caiete, care sunt elementele vectorului după fiecare etapă de execuție a algoritmului de selecție.
2. Modificați codul sursă al sortării prin selecție, astfel încât să fie ordonate descrescător doar valorile pare ale tabloului x .

Activitate remedială „MaxMat”

Item 5

Competențe specifice:

X.1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

X.2.1 Identificarea necesității structurării datelor în tablouri.

X.3.1 Prelucrarea datelor structurate

X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

Notă pentru profesori:

Activitatea propusă constă în două etape:

1. Etapa în care sunt dezvoltate competențele de prelucrare a datelor structurate și utilizarea unui mediu de programare la nivel 1 (cunoaștere). Elevul va recapitula operațiile de parcurgere și de referire la elementele unui tablou bidimensional.
2. Etapa constă într-un exercițiu remedial de același tip. În această etapă se pot aborda probleme disponibile pe platformele pbinfo.ro, campion.edu.ro, infoarena.ro etc.

Etapa 1 – recapitulare

Propunem activitățile remediale ”Organizarea datelor într-un tablou bidimensional” și/sau ”Completarea elementelor într-un tablou bidimensional”. Ambele activități pot fi realizate frontal, în clasă/laborator, dar și online.

Etapa 2 – exercițiu remedial de același tip**Enunț**

Se dă o matrice cu N linii și M coloane și elemente numere întregi.

Cerință

Să se determine, pentru fiecare linie a matricei, valoarea maximă și indicele coloanei pe care se află valoarea maximă.

Date de intrare

Fișierul de intrare `maxmat.in` conține pe prima linie numerele N și M, iar următoarele N linii câte M numere întregi, reprezentând elementele matricei.

Date de ieșire

Fișierul de ieșire `maxmat.out` va conține N linii; pe fiecare linie se vor afla două numere, separate prin spațiu, reprezentând valoarea maximă de pe linia curentă și indicele coloanei pe care se află aceasta.

Restricții și precizări

- $1 \leq N, M \leq 25$
- elementele matricei aparțin intervalului $[-1000, 1000]$
- dacă pe o linie a matricei, elementul maxim apare de mai multe ori, se va afișa indicele de coloană minim
- liniile și coloanele matricei sunt numerotate de la 1

(<https://www.pbinfo.ro/probleme/191/maxmat>)

Întrebare	Răspuns așteptat
Cum se poate scrie secvența de program care permite citirea datelor de intrare?	<pre>ifstream f("maxmat.in"); f>>N>>M; for(i=1; i<=N; i++) for(j=1; j<=M; j++) f>>v[i][j]; f.close();</pre>
Cum se poate scrie secvența de program care permite determinarea valorii maxime aflate pe o linie k?	<pre>max=v[k][1]; for(j=2; j<=M; j++) if(v[k][j]>max) max=v[k][j];</pre>
Cum se poate completa secvența anterioară astfel încât să reținem și poziția pe care se află valoarea maximă?	<pre>max=v[k][1]; poz=1; for(j=2; j<=M; j++) { if(v[k][j]>max) { max=v[k][j]; poz=j; } }</pre>

Activitate remedială „Interclasări”**Item 6****Competențe specifice:**

X.1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)

X.3.1 Prelucrarea datelor structurate

X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme

Notă pentru profesori:

Activitatea propusă constă în două etape:

1. Etapa premergătoare constă în reactualizarea cunoștințelor referitoare la o operație de prelucrare a datelor structurale de tip tablou unidimensional – interclasarea;

2. Exercițiul remedial de același tip

FIȘĂ DE LUCRU**Etapa 1**

1. Vizionați animația ce ilustrează pașii algoritmului de interclasare disponibil la adresa <https://www.pbinfo.ro/articole/5588/interclasarea-tablourilor>
2. Descrieți, în pseudocod, algoritmul de interclasare.

Etapa 2 – exercițiul remedial de același tip**Enunț**

Se dau două șiruri A și B, cu N, respectiv M elemente, numere naturale, ordonate strict crescător. Să se afișeze, în ordine strict crescătoare, valorile existente în ambele șiruri.

Date de intrare

Fișierul de intrare `interclasare2.in` conține pe prima linie numărul N; urmează N numere naturale, ordonate strict crescător, ce pot fi dispuse pe mai multe linii. Linia următoare conține numărul M și urmează M numere naturale, ordonate strict crescător, ce pot fi dispuse pe mai multe linii.

Date de ieșire

Fișierul de ieșire `interclasare2.out` va conține, în ordine strict crescătoare, valorile existente în ambele șiruri. Aceste valori vor fi afișate câte 10 pe o linie, separate prin spații. Ultima linie poate conține mai puțin de 10 de valori.

Restricții și precizări

$1 \leq N, M \leq 100000$

(<https://www.pbinfo.ro/probleme/251/interclasare2>)

Întrebare	Răspuns așteptat
Cum se poate scrie secvența de program care permite citirea datelor de intrare?	<pre>ifstream f("interclasare2.in"); f>>N; for (i=1; i<=N; i++) f>>A[i]; f>>M; for (j=1; j<=M; j++) f>>B[j]; f.close();</pre>
Cum se poate scrie secvența de program care permite interclasarea celor două șiruri și memorarea valorilor comune?	<pre>i=1, j=1; int k=0; while (i<=N && j<=M) if (A[i]<B[j]) i++; else if (A[i]>B[j]) j++; else { C[++k]=A[i++]; j++; }</pre>
Cum se poate scrie secvența de program care permite scrierea rezultatelor?	<pre>int cnt=0; ofstream g("interclasare2.out"); for (i=1; i<=k; i++) { g<<C[i]<<" "; cnt++; if (cnt%10==0) g<<'\n'; } g.close();</pre>

3.2 EXEMPLU DE ACTIVITATE DE RECUPERARE

Activitatea de recuperare „Frații”

Competențe specifice

- X.1.1 Implementarea algoritmilor reprezentați în pseudocod în limbaj de programare
- X.1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)
- X.2.1 Identificarea necesității structurării datelor în tablouri
- X.3.1 Prelucrarea datelor structurate
- X.2.3 Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor
- X.4.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării
- X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme
- XI.1.1 Analizarea unei probleme în scopul identificării datelor necesare și alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă
- XI.1.2 Identificarea avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme
- XI.1.3 Utilizarea șirurilor de caractere și a structurilor de date neomogene în modelarea unor situații problemă
- XI.1.4 Implementarea unor algoritmi de prelucrare a a șirurilor de caractere și a structurilor neomogene
- XI.3.1 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme
- XI.3.3 Utilizarea corectă a subprogramelelor predefinite și a celor definite de utilizator

Notă pentru profesori:

Recomandăm dezvoltarea competențelor specifice prin aplicații de sinteză pentru unitatea de învățare „Subprograme” și „Tipuri structurate de date” din programa clasei a XI-a.

Activitatea are mai multe sarcini de lucru, gradate ca nivel de dificultate: primele două sarcini de lucru sunt de nivel 1, următoarele două sunt de nivel 2, iar ultima sarcină de lucru este de nivel 3.

Modul de organizare a activității la clasă este la decizia profesorului, în funcție de specificul clasei. O posibilitate de abordare ar fi ca primele 3 sarcini de lucru să fie rezolvate în clasă, individual de către fiecare elev, timpul de lucru fiind specificat de către profesor; la expirarea timpului de lucru (sau mai repede dacă toți elevii au terminat), un elev prezintă modul de rezolvare; elevii sunt solicitați să exprime opinii referitoare la corectitudinea/eficiența rezolvării și să prezinte și alte abordări (dacă există). Profesorul poate prezenta la final o altă modalitate de implementare, dacă soluțiile elevilor nu au fost optime. La finalul acestei etape a activității toți elevii clasei vor avea implementate versiuni corecte și eficiente pentru subprogramele solicitate (cu sprijinul profesorului, dacă este necesar).

Se recomandă ca elevii să transmită rezolvările realizate către profesor (recomandabil pe platforma utilizată pentru organizarea activității la clasă sau prin e-mail). Profesorul testează și analizează fiecare rezolvare și oferă feed-back elevilor, oferind acestora posibilitatea de a trimite o nouă variantă, îmbunătățită, care să valorifice feed-back-ul oferit de către profesor.

FIȘĂ DE LUCRU

1. Efectuați pe caiet următoarele operații și transformați rezultatul în fracție ireductibilă:

$$\frac{2}{14} + \frac{5}{21}$$

$$\frac{1}{20} - \frac{13}{22}$$

$$\frac{15}{20} \cdot \frac{2}{5}$$

$$\frac{65}{12} \cdot \frac{13}{22}$$

Exprimați în limbaj natural procedeul aplicat pentru efectuarea operației. Identificați secvența de pași executată pentru efectuarea operației.

2. Proiectați o structură de date capabilă să rețină numitorul și numărătorul unei fracții (numere întregi $< 10^9$). Declarați tipul de date corespunzător.

3. Definiți următoarele subprograme:

- i. subprogramul `simplifica` care primește ca parametru o fracție f și transformă fracția în fracție ireductibilă;
- j. subprogramul `adun` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 suma fracțiilor f_1 și f_2 în formă ireductibilă;
- k. subprogramul `scad` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 rezultatul scăderii fracției f_2 din fracția f_1 în formă ireductibilă;
- l. subprogramul `produs` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 produsul fracțiilor f_1 și f_2 în formă ireductibilă;
- m. subprogramul `impart` care primește ca parametri două fracții f_1 și f_2 și returnează prin intermediul celui de al treilea parametru f_3 rezultatul împărțirii fracției f_1 la fracția f_2 în formă ireductibilă;
- n. subprogramul `compar` care primește ca parametri fracțiile f_1 și f_2 și returnează ca rezultat valoarea $-1, 0$ sau 1 după cum fracția f_1 este mai mică, egală, respectiv mai mare decât fracția f_2 ;
- o. subprogramul `sortare` care primește ca parametri un număr natural n și un vector F cu n fracții, memorate pe pozițiile $0, 1, \dots, n-1$ și sortează crescător fracțiile din vectorul F ;
- p. subprogramul `scrie` care primește ca parametru o fracție f și o afișează pe ecran în forma numitor/numărător.

Evaluarea activității

Ponderea punctajelor în evaluarea activității

Cerința	1	2	3
Pondere	5%	5%	$8 \cdot 10\% = 80\%$

Pentru fiecare criteriu sunt prezentați descriptori pentru 4 niveluri de performanță, modul de alocare a punctajelor fiind sugerat în paranteză, dar rămânând la decizia profesorului.

Cerința 1:

Criteriu	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Rezolvarea operațiilor	Rezultatul este corect pentru fiecare operație. Modul de rezolvare ilustrează pașii urmați.	Rezultatul este corect pentru fiecare operație. Nu este clar ilustrat modul în care a fost obținut rezultatul respectiv.	Cel puțin 3 dintre operațiile specificate au rezultat corect.	Operațiile nu sunt executate corect.
2. Exprimarea în limbaj natural cu evidențierea secvenței de pași.	Exprimarea este fluentă și corectă. Elevul se exprimă cu ușurință în limbaj științific. Pașii urmați pentru efectuarea fiecărei operații sunt clar specificați, secvențierea fiind corectă.	Exprimarea este corectă, dar elevul are dificultăți de a transpune cu claritate în cuvinte procedeul aplicat, cel puțin pentru una dintre operații. Pașii urmați sunt specificați cu claritate, poate nu în ordinea strictă a executării lor.	Exprimarea este sumară, termenii utilizați nu sunt științifici, pașii nu sunt delimitați cu claritate, nici secvențierea nu este corectă pentru cel puțin două dintre operații.	Elevul nu poate descrie în cuvinte procedeul folosit pentru cel puțin 3 dintre operații. Nu identifică pașii care trebuie urmați pentru efectuarea operațiilor.

Cerința 2

Criteriu	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Proiectarea structurii de date	Elevul identifică necesitatea utilizării unei structuri de date care să rețină atât	Elevul identifică necesitatea utilizării unei structuri de date care să rețină atât	Elevul identifică necesitatea utilizării unei structuri de date care să	Elevul nu identifică necesitatea utilizării unei structuri de date care să rețină atât

	numitorul, cât și numărătorul fracției. Alege tipul struct/record drept cea mai bună modalitate de implementare a acestei structuri. Declară corect din punct de vedere sintactic tipul de date corespunzător.	numitorul, cât și numărătorul fracției. Alege tipul struct/record drept cea mai bună modalitate de implementare a acestei structuri. Are dificultăți în a declara corect din punct de vedere sintactic tipul de date corespunzător.	reține atât numitorul, cât și numărătorul fracției. Nu identifică tipul struct/record drept varianta optimă de implementare. Declară corect din punct de vedere sintactic o altă modalitate de implementare.	numitorul, cât și numărătorul fracției. Declarațiile sunt pentru variabile independente pentru numitor/numărător.
--	--	---	--	---

Cerința 3

Pentru fiecare dintre funcții se aplică următoarele criterii:

Criteria	4 (10-9)	3 (8-7)	2 (6-5)	1 (4-3-2-1)
1. Antetul funcției	Antetul funcției este corect din punct de vedere sintactic. Funcția are parametri specificați. Declararea parametrilor permite transferul rezultatului calculat de funcție.	Antetul funcției este corect din punct de vedere sintactic. Funcția are parametri specificați. Declararea parametrilor nu permite transferul rezultatului calculat de funcție.	Antetul funcției este corect din punct de vedere sintactic. Funcția nu are parametri specificați. Declararea parametrilor nu permite transferul rezultatului calculat de funcție.	Antetul funcției nu este corect.
2. Implementarea operației	Subprogramul implementează corect operația. Rezultatul este transferat corect, prin intermediul parametrului specificat.	Subprogramul implementează corect operația. Rezultatul nu este transferat corect, prin intermediul parametrului specificat.	Subprogramul implementează principial corect operația, dar nu funcționează pentru orice set de date.	Subprogramul nu implementează corect operația specificată.

Secțiunea 4 – Adaptarea la particularitățile/categoriile de elevi în risc. Exemple de activități de învățare

Activitatea de învățare „Declararea tablourilor unidimensionale“

Notă pentru profesori:

Activitatea este structurată ca o conversație euristică, având scopul de a descoperi prin întrebări succesive necesitatea utilizării tablourilor unidimensionale, precum și modul corect de declarare a acestora. Activitatea poate fi parcursă parțial sau integral, în funcție de unitatea de învățare în care este abordată (Șiruri de caractere/Structuri de date neomogene/Subprograme).

FIȘĂ DE LUCRU 1

Competența specifică:

X.3.1 Prelucrarea datelor structurate în tablouri.

Cele 18 medii semestriale la disciplinele studiate și media aritmetică a acestora sunt memorate într-o variabilă. Cum ați declara corect această variabilă?

- | | |
|-------------------------------|---------------------------------|
| a. <code>double m[18];</code> | b. <code>float m[19];</code> |
| c. <code>int m[19];</code> | d. <code>long int m[20];</code> |

Pentru următoarea declarație generică a unui tablou unidimensional (vector):

```
tip nume[NRMAX];
```

Întrebare		Răspuns așteptat
Câte elemente are tabloul declarat?		NRMAX
Cum sunt numerotate/indexate elementele din tabloul declarat?		0, 1, ... NRMAX-1
Care este dimensiunea zonei de memorie alocate elementelor tabloului?		NRMAX*sizeof(tip)
Câte valori avem de memorat în vector și ce tip trebuie să alegem pentru elementele acestuia?		18+1 valori Cele 18 medii semestriale/disciplină sunt numere naturale. Media semestrială a elevului este număr real. → Alegem tipul cel mai cuprinzător, adică un tip real.
Care este numărul maxim de elemente pe care le poate memora fiecare dintre tablourile declarate anterior și care este tipul elementelor în fiecare caz?	a. <code>double m[18];</code>	18 elemente numere reale
	b. <code>float m[19];</code>	19 elemente numere reale → răspuns corect
	c. <code>int m[19];</code>	19 elemente numere întregi
	d. <code>long int m[20];</code>	20 elemente numere întregi
În scop didactic, utilizăm uneori numerotarea/indexarea elementelor de la valoarea 1. Având în vedere cerința problemei, cum ar trebui declarată variabila în acest caz?		<code>float m[20];</code>

Activitatea de învățare „Ghicește numărul”

Notă pentru profesori:

În cazul în care se constată la unii elevi carențe în dezvoltarea competențelor anterioare, se recomandă dezvoltarea acestora în aplicații pentru unitatile de învățare Liste/ Subprograme, din programa clasei a XI-a. Activitatea de recuperare își propune să dezvolte la elevi deprinderea de a folosi eficient algoritmi de prelucrare a tablourilor unidimensionale.

Competențe specifice:

X.3.1. Prelucrarea datelor structurate în tablouri.

X.4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme.

FIȘĂ DE LUCRU 1

„Ghicește numărul” este un joc în care trebuie ghicit un număr secret format din două cifre, selectat de către calculator. O încercare e formată din numărul propus, ales de tine, și răspunsul calculatorului (**Ai ghicit!** / **Prea mare!** / **Prea mic!**) Utilizând informația din răspunsul calculatorului, numărul trebuie ghicit folosind cât mai puține încercări posibile.

Întrebare	Răspuns așteptat
Care este situația cea mai defavorabilă?	Încerc toate cele 90 variante de răspuns.
Cum ați parcurge șirul de numere?	Secvențial.
Dar dacă calculatorul vă oferă un răspuns de tipul: Ai ghicit! / Prea mare! / Prea mic!	Încerc elementul din mijlocul șirului dat: șir: 10, 11, 12, ..., 45 , ..., 99 ↑ ↑ ↑ poziție: 1 mijloc 90 Dacă nu am ”noroc”, în funcție de răspunsul calculatorului caut în subșirul din stânga / din dreapta mijlocului. Continui același procedeu cu noul subșir până la ghicirea numărului.
Vizionați videoclipul Căutare binară - dans Flamenco https://www.youtube.com/watch?v=iP897Z5Nerk și determinați câte comparații au fost necesare până la găsirea numărului aflat pe costumul băiatului.	3 comparații (cu valorile 5, 8 și 7)
Urmăriți animațiile de la adresa de mai jos, care ilustrează o comparație între cele două tipuri de căutări: <ul style="list-style-type: none"> • secvențială, varianta I. • binară, varianta II. https://blog.penjee.com/binary-vs-linear-search-animated-gifs/ <p>Care dintre cei doi algoritmi este mai eficient ca timp de execuție?</p>	Căutarea binară.
Pot aplica căutarea binară oricărui șir de numere?	Doar celor ordonate.

Fișă de lucru 2

Se dau trei secvențe de numere:

- i. 2, 5, 6, 9, 8, 10, 16
- ii. 9, 7, 6, 3, 2, 1
- iii. 2, 4, 5, 7, 12, 13, 15

a. Cărei secvențe i se poate aplica căutarea binară fără alte prelucrări prealabile?

Întrebare	Răspuns așteptat	
Analizați variantele de răspuns și precizați motivul pentru care nu putem aplica căutarea binară.	i. 2, 5, 6, 9, 8, 10, 16	Nu se poate aplica pentru că șirul nu este ordonat.
	ii. 9, 7, 6, 3, 2, 1	Se poate aplica.
	iii. 2, 4, 5, 7, 12, 13, 15	Se poate aplica.

b. În secvența iii. pentru a afla poziția pe care se află valoarea $x=12$, se aplică metoda căutării binare. Care este succesiune a corectă a elementelor a căror valoare se compară cu elementul x ?

Întrebare	Răspuns așteptat
Cum determin poziția elementului din mijloc?	$mij = (st + dr) / 2;$
Cum se procedează în continuare ?	<p>Se compară x cu elementul aflat la mijlocul șirului.</p> <ul style="list-style-type: none"> În cazul în care cele două elemente coincid, căutarea s-a încheiat cu succes. Dacă x este mai mare decât elementul din mijlocul șirului, se continuă căutarea în aceeași manieră în subșirul delimitat de elementul din mijloc și finalul șirului inițial. Dacă x este mai mic decât elementul din mijloc, se continuă căutarea în aceeași manieră în subșirul delimitat de elementul din mijloc și începutul șirului inițial. <p>$mij = (1+7)/2=4$</p> <p>șir: 2, 4, 5, 7, 12, 13, 15 ↑ ↑ ↑ poziție: 1 mij 7</p> <p>12 este mai mare decât elementul din mijloc → caut în subșirul din dreapta mijlocului</p> <p>$mij = (5+7)/2=6$</p> <p>șir: 12, 13, 15 ↑ ↑ ↑ poziție: 5 mij 7</p> <p>12 este mai mic decât elementul din mijloc → caut în subșirul din stânga mijlocului</p> <p>$mij = (5+5)/2=5$</p> <p>șir: 12 ↑ poziție: 5 = mij</p> <p>12 este egal cu elementul din mijloc și poziția pe care se află în șir este 5. răspuns corect: 7, 13, 12</p>
Ce se schimbă în algoritm dacă șirul inițial este ordonat descrescător? 15, 13, 12, 7, 5, 4, 2	<ul style="list-style-type: none"> Dacă x este mai mare decât elementul din mijlocul șirului, se continuă căutarea în aceeași manieră în subșirul din stânga elementului din mijloc. Dacă x este mai mic decât elementul din mijloc, se continuă căutarea în aceeași manieră în subșirul din dreapta elementului din mijloc.

Activitate de învățare „Interclasarea“

Competențe specifice:

X.3.1 Prelucrarea datelor structurate în tablouri.

X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme.

Notă pentru profesori:

Activitatea de recuperare își propune să dezvolte la elevi deprinderea de a folosi algoritmi eficienți din punct de vedere al timpului de executare pentru prelucrarea tablourilor unidimensionale. În cazul în care se constată la unii elevi carențe în dezvoltarea competențelor anterioare, se recomandă dezvoltarea acestora în aplicații pentru unitatile de învățare Liste/ Subprograme, din programa clasei a XI-a.

FIȘĂ DE LUCRU

Se consideră două tablouri unidimensionale

$a = (1, 3, 5, 9, 10)$ cu $n=5$ elemente și

$b = (2, 4, 6, 7)$ cu $m=4$ elemente.

a. Precizați care este valoarea elementului $a[b[0]]$.

b. În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:

i. (1,3,5,9,10,2,4,6,7)

ii. (1,2,3,4,5,6,9,7,10)

iii. nu se poate realiza interclasarea

iv. (1,2,3,4,5,6,7,9,10)

c. Care este complexitatea timp a algoritmului de interclasare?

Întrebare	Răspuns așteptat	
Din cerință se deduce ca vectorii sunt indexați de la 0. Care este valoarea lui $b[0]$? Ce valoare are $a[b[0]]$?	$b[0]=2$ $a[b[0]]=a[2]=5$	
Urmăriți modul de funcționare a algoritmului de sortare prin interclasare: https://eduonline.ro/edu.net/repository/pl-tic-7-2-2%20lectii/M4/index.html Ce condiție trebuie să îndeplinească tablourile inițiale?	Să fie ordonate → varianta iii. este incorectă	
Câte elemente va conține tabloul final ?	$m+n=9$ elemente	
Analizați variantele de răspuns rămase.	i. (1,3,5,9,10,2,4,6,7)	tabloul nu e sortat → răspuns incorect
	ii. (1,2,3,4,5,6,9,7,10)	tabloul nu e sortat → răspuns incorect
	iv. (1,2,3,4,5,6,7,9,10)	→ răspuns corect
Care este complexitatea timp a algoritmului de interclasare ?	Complexitatea timp a algoritmului este $O(m+n)$ deoarece fiecare element din a și b a fost folosit o singură dată.	

Activitate de învățare „Prelucrări în matrice pătratice“

Competențe specifice:

X.3.1 Prelucrarea datelor structurate în tablouri.

X.4.2 Alegerea unui algoritm eficient de rezolvare a unei probleme.

Notă pentru profesori:

Activitatea de recuperare își propune să dezvolte la elevi deprinderea de a folosi eficient algoritmi de prelucrare a tablourilor bidimensionale. În cazul în care se constată la unii elevi carențe în dezvoltarea competențelor anterioare, se recomandă dezvoltarea acestora în aplicații pentru unitatile de învățare Subprograme, din programa clasei a XI-a.

FIȘĂ DE LUCRU

În următoarea secvență de program, a este o matrice cu n linii și n coloane având elemente întregi; liniile și coloanele matricei a sunt numerotate de la 1 la n . Variabilele i , j și s sunt de tip întreg.

```
s=0;
i=1;
while (i<=n)
{
  j=n;
  while (j>=1)
  {
    if (i==j)
      s = s + a[i][j];
    j--;
  }
}
```

Stabiliți ce reprezintă valoarea memorată în variabila s la finalul execuției algoritmului și care este complexitatea algoritmului.

- suma elementelor de pe diagonala principală / $O(n^2)$
- suma elementelor de pe diagonala principală / $O(n)$
- suma elementelor de pe diagonala secundară / $O(n)$
- suma elementelor de pe diagonala principală / $O(2n)$

Activitatea poate fi concepută ca o conversație euristică în care, prin întrebări succesive, profesorul ghidează elevul către descoperirea răspunsului corect.

Întrebare	Răspuns așteptat
Cum identificăm elementele aflate pe diagonala principală într-o matrice pătratică de dimensiune n , indexată de la 1?	Indicele de linie este egal cu indicele de coloană.
Cum identificăm elementele aflate pe diagonala secundară într-o matrice pătratică de dimensiune n , indexată de la 1?	Suma dintre indicele de linie și indicele de coloană este $n+1$ → punctul c este incorect
Ce valori iau variabilele i , respectiv j în secvența anterioară? Ce complexitate deduceți că are algoritmul?	$i \in \{1, 2, 3, \dots, n\}$ $j \in \{n, n-1, n-2, \dots, 1\}$ $O(n^2)$ → răspuns corect a
Cum s-ar putea crește eficiența timp a acestui algoritm? Care va fi complexitatea timp în acest caz?	Dat fiind că $i=j$, putem folosi o singură instrucțiune repetitivă pentru calculul sumei elementelor de pe diagonala principală. $s=0;$ for ($i=1; i \leq n; i++$) $s=s+a[i][i];$ $O(n)$
Modificați secvența anterioară astfel încât să calculeze suma elementelor de pe diagonala secundară a unei matrice pătratică de dimensiune n , indexată de la 1?	$i+j=n+1$, $1 \leq i \leq n$, $1 \leq j \leq n$ → $j=n+1-i$ $s=0;$ for ($i=1; i \leq n; i++$) $s=s+a[i][n+1-i];$
Dar dacă matricea ar fi indexată de la 0?	Condiția ca un element să se găsească pe diagonala secundară într-o matrice pătratică de dimensiune n , indexată de la 0 este: $i+j=n-1$, $0 \leq i < n$, $0 \leq j < n$ → $j=n-1-i$ $s=0;$ for ($i=0; i < n; i++$) $s=s+a[i][n-1-i];$

Activitate de învățare „Lucrul cu fișiere text“

Competențe specifice:

X.2.1 Identificarea necesității structurării datelor în tablouri

X.3.1 Prelucrarea datelor structurate în tablouri.

X.2.3 Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor

Notă pentru profesori:

Activitatea propusă constă în două etape:

1. reactualizarea cunoștințelor referitoare la lucrul cu fișiere text;
2. activitate de recuperare de același tip

FIȘĂ DE LUCRU

Etapa 1 – reactualizare

Urmăriți videoclipul aflat la adresa <https://www.youtube.com/watch?v=i3SqxwmGwOw> și notați în caiete operațiile specifice lucrului cu fișiere text identificate în videoclip. Marcați cu o culoare acele operații specifice lucrului cu fișiere text pe care vi le amintiți.

Comparați notițele voastre cu operațiile specifice lucrului cu fișiere text din articolul aflat la adresa: <https://www.pbinfo.ro/articole/19047/operatii-de-intrare-iesire-cu-fisiere-in-cpp>.

Etapa 2 – activitate de recuperare de același tip

Se dau n numere naturale cu cel mult două cifre fiecare. Să se determine acele numere care apar o singură dată.

Date de intrare

Fișierul de intrare `unice.in` conține pe prima linie numărul n ; urmează n numere naturale cu cel mult două cifre fiecare, dispuse pe mai multe linii și separate prin spații.

Date de ieșire

Fișierul de ieșire `unice.out` va conține pe prima linie, în ordine crescătoare, valorile care apar o singură dată, separate printr-un spațiu.

Restricții și precizări

$$1 \leq n \leq 100000$$

Exemplu

```
unice.in
7
3 5 2 1
5 23 1
```

```
unice.out
2 3 23
```

(<https://www.pbinfo.ro/probleme/267/unice>)

Întrebare	Răspuns așteptat
Ce bibliotecă este necesar să includem pentru lucrul cu fișiere text?	<code>#include <fstream></code>
Cum vom memora numărul de apariții ale fiecărei valori?	Vom folosi un vector de frecvență cu 100 de elemente. <code>v[x]</code> va reține numărul de apariții ale valorii x (citită din fișierul de intrare).
Cum se pot declara și deschide fișierele text necesare pentru rezolvarea problemei?	<code>ifstream fin("unice.in");</code> <code>ofstream fout("unice.out");</code>
Cum se poate scrie secvența de program care citește datele de intrare și construiește vectorul de frecvență?	<code>fin>>n;</code> <code>i=1;</code> <code>while (i<=n)</code> <code>{fin>>x;</code> <code> v[x]++;</code> <code> i++;</code> <code>}</code>

Cum identificăm o valoare care a apărut o singură dată în fișierul de intrare?	Dacă în vectorul de frecvență pe o poziție x găsim valoarea 1, înseamnă că valoarea x care a apărut o singură dată în fișierul de intrare
Cum se poate scrie secvența de program care scrie rezultatele în fișierul de ieșire?	<pre>for(i=0; i<100; i++) if(v[i]==1) fout<<i<<" "; fout.close();</pre>

Activitate de învățare „Algoritmi fundamentali de prelucrare a datelor structurate în tablouri unidimensionale“

Competențe specifice:

X.2.1 Identificarea necesității structurării datelor în tablouri

X.3.1 Prelucrarea datelor structurate în tablouri.

Notă pentru profesori:

Activități propuse:

➤ lecții AEL din setul:

- Tablouri (vectori, matrice, șiruri de caractere) <https://eduonline.roedu.net/repository/pl-tic-7-1-2%20lectii/>

Tablouri (vectori, matrice, șiruri de caractere)
Inserarea și ștergerea unei componente

Teorie

Pentru a insera o valoare se procedează diferentiat astfel:

a) Dacă inserarea se face după ultima componentă, atunci ea este imediată, dar variabila care reține numărul componentelor vectorului trebuie incrementată (să i se adune 1).

b) Dacă inserarea se face în interior, atunci, toate valorile care trebuie să se găsească după cea care se inserează se deplasează către dreapta cu o poziție. Și în acest caz trebuie să se incrementeze variabila care reține numărul de componente ale vectorului.

Analizați câte un exemplu pentru fiecare caz:

a) b)

Simulare

Cazul b). Inserare în interior, în poziția a treia

V[7] 45 12 129 7 ? ? ?
 1 2 3 4 5 6 7

n 5

algoritmul în pseudocod

```
n ← n + 1
i ← n
pos ← 3
cat timp (i#pos) execută
| V[i] ← V[i-1]
| i ← i - 1
V[pos] ← 82
```

- Algoritmi elementari de lucru cu tablourile <https://eduonline.roedu.net/repository/pl-tic-7-2-2%20lectii/>

Algoritmi elementari de lucru cu tablourile
Algoritm de interclasare

Teorie

Se consideră doi vectori cu numere naturale A și B. În fiecare vector numerele sunt sortate crescător. Se cere să se obțină, într-un vector C, în ordine crescătoare, toate valorile reținute de A și B.

Ideea algoritmului: se pornește cu valorile din primele componente ale vectorilor A și B. Cea mai mică dintre ele se adaugă în C. În vectorul a cărui valoare a fost trecută în C se trece la componenta următoare. Se continuă în acest mod până când au fost trecute în C toate valorile dintr-un vector. Valorile rămase în celălalt vector se trec, în ordine, în C.

Pornește animația alăturată!

Simulare

a[4] 10 14 21 38
 1 2 3 4

b[3] 2 39 56
 1 2 3

c[7] 2 10 ? ? ? ? ?
 1 2 3 4 5 6 7

Gândirea algoritmică

- Captarea atenției
- 1. Noțiunea de algoritm
- 2. Exemple de algoritmi
- 3. Evaluare (1)
- 4. Descrierea algoritmilor
- 5. Exerciții
- 6. Evaluare (2)
- 7. Structuri (Partea I)
- 8. Structura liniară (exemple și exerciții)
- 9. Evaluare (3)
- 10. Structuri (Partea a II-a)
- 11. Structura alternativă (exemple și exerciții)
- 12. Evaluare (4)
- 13. Structuri (Partea a III-a)
- 14. Structura repetitivă (exemple și exerciții)
- 15. Evaluare (5)



Gândirea algoritmică

Structura alternativă (exemple și exerciții)



Built to teach, not to gentry

Cerință

Fie trei localități: A, B, C și:

- a - distanța de la A la B;
- b - distanța de la B la C;
- c - distanța de la C la A.

Vizitează toate cele trei orașe, pornind din oricare dintre orașele A sau B, ajungând în final, în oricare dintre orașele B sau C, alegând un traseu de lungime minimă.

Distanțe



Ruta:
Distanța parcursă: 0

➤ Soft educațional propus de prof. Țopa Robert
<http://www.cnchogastecuci.ro/portofoliu/vectori/index.html>

Tablouri unidimensionale - Vectori

NOȚIUNI TEORETICE
BUBBLE SORT
SELECȚIA DIRECTĂ
PROBLEME PROPUSE

Probleme rezolvate

Inserarea unui element într-un vector - joc + test de evaluare

Aranjarea elementelor într-un vector după anumite criterii - problemă rezolvată

Folosirea unui vector caracteristic pentru ordonarea și pentru contorizarea unor numere - problemă rezolvată

Căutarea unei poziții și inserarea unui element în acea poziție - problemă rezolvată

Test de evaluare - verificarea cunoștințelor

Bubble Sort - mecanism de execuție

Click pe săgeată pentru a urmări mecanismul de execuție al algoritmului. Secvența de instrucțiuni C++ pentru această metodă este mai jos.

prima parcurgere gata=1

i=0	4	3	2	1
-----	---	---	---	---

interschimbare, gata=0

i=1	3	4	2	1
-----	---	---	---	---

interschimbare, gata=0

▶

```

.....
while (gata==0)
{
    gata=1;
    for (i=0; i<n-1; i++)
        if (v[i]>v[i+1])
        {
            x=v[i]; v[i]=v[i+1];
            v[i+1]=x; gata=0;
        }
}
.....

```

➤ Bubble Sort – dans popular ungar <https://www.youtube.com/watch?v=lyZQPjUT5B4&t=252s>

125

ANEXE

Anexa 1

În tabelul de mai jos sunt prezentate competențele și sunt reținute și conținuturile din programa clasei a IX-a Matematică-Informatică și Științe ale Naturii, care sunt corelate cu competențele și conținuturile clasei a X-a, ce urmează a fi atinse în anul școlar 2020-2021.

Clasa a IX-a		Clasa a X-a	
Competențe specifice 2019-2020	Conținuturi	Competențe specifice aflate în corespondență în 2020-2021	Conținuturi
<p>3.1. Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.</p> <p>3.2. Reprezentarea algoritmilor în pseudocod</p> <p>3.3. Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor</p>	<p>Algoritmi elementari Prelucrarea numerelor :</p> <ul style="list-style-type: none"> • prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietății de palindrom etc.) • probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate etc.) • calculul unor expresii simple (sume, produse etc.) <p>Prelucrarea unor secvențe de valori</p> <ul style="list-style-type: none"> • determinare minim/maxim • verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte etc.) • calculul unor expresii în care intervin valori din secvență (de exemplu: numărarea elementelor pare/impare etc) • generarea șirurilor recurente (de exemplu: șirul Fibonacci) 	<p>1.1 Implementarea algoritmilor reprezentați în pseudocod în limbaj de programare</p> <p>1.2 Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)</p>	<p>Elementele de bază ale limbajului de programare Noțiuni introductive</p> <ul style="list-style-type: none"> • Structura programelor • Vocabularul limbajului • Tipuri simple de date (standard) • Constante, variabile, expresii • Citirea/scrierea datelor <p>Structuri de control</p> <ul style="list-style-type: none"> • Structura liniară • Structura alternativă • Structuri repetitive <p>Mediul limbajului de programare studiat</p> <ul style="list-style-type: none"> • Prezentare generală • Editarea programelor sursă • Compilare, rulare, depanare
<p>Aplicații interdisciplinare</p> <p>4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării</p> <p>4.2. Alegerea celui mai eficient algoritm de rezolvare a unei probleme</p> <p>5.1. Identificarea într-un program a structurilor de control învățate</p>	<p>Aplicații interdisciplinare (specifice profilului)</p> <ul style="list-style-type: none"> • Rezolvarea ecuației de gradul I și de gradul al II-lea • Simplificarea fracțiilor • Aplicații geometrice (distanța dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate etc.) • Determinarea punctului de intersecție a două mobile în mișcare rectilinie și uniformă • Determinarea masei moleculare a unui compus chimic. <p>Analiza eficienței unui algoritm. Exemplificări de modalități de implementare a algoritmilor studiați.</p>		<p>Implementarea unor algoritmi elementari cu aplicabilitate practică</p>

Anexa 2

În tabelul de mai jos sunt prezentate competențele și sunt reținute și conținuturile din programa clasei a X-a Matematică-Informatică, care sunt corelate cu competențele și conținuturile clasei a XI-a, ce urmează a fi atinse în anul școlar 2020-2021.

Clasa a X-a		Clasa a XI-a	
Competențe specifice 2019-2020	Conținuturi	Competențe specifice aflate în corespondență în 2020-2021	Conținuturi
2.1. Identificarea necesității structurării datelor în tablouri 2.2. Prelucrarea datelor structurate în tablouri./ 3.1. Prelucrarea datelor structurate în tablouri.	Tipuri structurate de date Tipul tablou. Tablouri unidimensionale și bidimensionale. Algoritmi fundamentali de prelucrare a datelor structurate în tablouri <ul style="list-style-type: none"> căutare secvențială, căutare binară sortare interclasare prelucrări specifice tablourilor bidimensionale 	1.1. Analizarea unei probleme în scopul identificării datelor necesare și alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă 1.2. Identificarea avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme 1.3. Utilizarea șirurilor de caractere și a structurilor de date neomogene în modelarea unor situații problemă 1.4. Implementarea unor algoritmi de prelucrare a a șirurilor de caractere și a structurilor neomogene	Tipuri structurate de date Șiruri de caractere <ul style="list-style-type: none"> Particularități de memorare a șirurilor de caractere Subprograme predefinite de prelucrare a șirurilor de caractere Structuri de date neomogene (struct/record) <ul style="list-style-type: none"> Rezolvarea unor probleme cu caracter practic. Liste <ul style="list-style-type: none"> Reprezentarea grafică a structurilor de tip listă Operații specifice Stiva și coada Aplicații cu implementare statică
4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme	Aplicații interdisciplinare Exemple orientative: <ul style="list-style-type: none"> Prelucrări statistice ale unei serii de valori Calculul valorii unei expresii algebrice Calcul combinatoriale Determinarea unor mărimi fizice dintr-un circuit electric Aplicații din genetică (legea creșterilor organice, etc.) Analiza eficienței unui algoritm	3.1 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme 3.2 Analiza problemei în scopul identificării subproblemelor acesteia 3.3 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator	Subprograme <ul style="list-style-type: none"> Structura și a modul de definire al subprogramelor Declararea și apelul subprogramelor Transferul parametrilor la apel (prin valoare și referință) Returnarea valorilor de către subprograme Variabile locale și globale Aplicații folosind subprograme
5.1. Identificarea aplicațiilor informaticii în viața socială 5.2. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	Aplicații din viața cotidiană Exemple orientative: <ul style="list-style-type: none"> Determinarea situației școlare a unui elev (medii absențe, etc.) Balanța de cheltuieli ale unei familii Determinarea salariului unei persoane Evidența operațiilor într-un cont bancar 		

Anexa 3

În tabelul de mai jos sunt prezentate competențele și sunt reținute și conținuturile din programa clasei a IX-a Matematică-Informatică, intensiv informatică, care sunt corelate cu competențele și conținuturile clasei a X-a, ce urmează a fi atinse în anul școlar 2020-2021.

Clasa a IX-a		Clasa a X-a	
Competențe specifice 2019-2020	Conținuturi	Competențe specifice aflate în corespondență în 2020-2021	Conținuturi
4.2. Identificarea necesității structurării datelor în tablouri. 4.3. Prelucrarea datelor structurate 4.5. Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)	<p>Tipuri structurate de date. Tipul tablou</p> <ul style="list-style-type: none"> • Tablouri unidimensionale • Tablouri bidimensionale <p>Algoritmi fundamentali de prelucrare a datelor structurate în tablouri</p> <ul style="list-style-type: none"> • căutare secvențială, căutare binară • sortare • interclasare • parcurgerea tablourilor bidimensionale pe linii/coloane 	<p>1.1. Evidențierea necesității structurării datelor. 1.2. Prelucrarea datelor structurate. 1.3. Alegerea structurii de date adecvate rezolvării unei probleme. 3.1. Prelucrarea datelor structurate.</p>	<p>Tipuri structurate de date</p> <ul style="list-style-type: none"> • Șir de caractere • Înregistrare (structură) • Listă, stivă, coadă <p>Structuri de date alocate dinamic (definiții, utilitate)</p> <ul style="list-style-type: none"> • Liste simplu înlănțuite • Liste dublu înlănțuite • Liste circulare • Operații elementare pe liste înlănțuite (inserare element, ștergere element, parcurgere)
5.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 5.2. Alegerea unui algoritm eficient de rezolvare a unei probleme	<p>Aplicații interdisciplinare (specifice profilului) Exemple orientative:</p> <ul style="list-style-type: none"> • Rezolvarea ecuației de gradul I și de gradul al II-lea • Simplificarea fracțiilor • Aplicații geometrice (distanța dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate, centrul de greutate al unei mulțimi de puncte, etc.) • Determinarea primilor n termeni ai unei progresii aritmetice/geometrice • Determinarea punctului de intersecție a două mobile în mișcare rectilinie și uniformă • Determinarea masei moleculare a unui compus chimic <p>Analiza eficienței unui algoritm.</p>	<p>5.1. Identificarea aplicațiilor informaticii în viața socială. 5.2. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene</p>	<p>Aplicații din viața cotidiană. Exemple orientative:</p> <ul style="list-style-type: none"> • Determinarea situației școlare a unei clase de elevi (medii semestriale, medii generale, numărul de absențe, etc.) • Evidența cheltuielilor la întreținere pentru locatarii unui bloc • Evidența operațiilor cu conturi bancare.
		<p>4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme</p>	<p>Aplicații interdisciplinare (specifice profilului) cu structuri de date și subprograme. Exemple orientative:</p> <ul style="list-style-type: none"> • Prelucrări statistice ale unei serii de valori • Calcule și generări combinatoriale • Determinarea unor mărimi fizice dintr-un circuit electric serie/ paralel • Aplicații din genetică (legea creșterilor organice, etc) • Prelucrarea de text (căutarea unui subșir într-un șir, transformări, delimitări cuvinte, etc) <p>Analiza eficienței unui algoritm</p>

Anexa 4

În tabelul de mai jos sunt prezentate competențele și sunt reținute și conținuturile din programa clasei a X-a Matematică-Informatică – intensiv informatică, care sunt corelate cu competențele și conținuturile clasei a XI-a, ce urmează a fi atinse în anul școlar 2020-2021.

Clasa a X-a		Clasa a XI-a	
Competențe specifice 2019-2020	Unități de învățare/Conținuturi	Competențe specifice aflate în corespondență în 2020-2021	Unități de învățare/Conținuturi
1.1. Evidențierea necesității structurării datelor 1.2. Prelucrarea datelor structurate 1.3. Alegerea structurii de date adecvate rezolvării unei probleme 3.1. Prelucrarea datelor structurate	Structuri de date alocate dinamic Alocarea dinamică a memoriei (operații și mecanisme specifice) Structuri de date alocate dinamic (definiții, utilitate) <ul style="list-style-type: none"> • Liste simplu înlănțuite • Liste particulare (stiva, coada) • Liste dublu înlănțuite • Liste circulare • Operații elementare pe liste înlănțuite (inserare element, ștergere element, parcurgere, căutarea unui element într-o listă) 	1.2. Analizarea unei probleme în scopul identificării datelor necesare și alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă 1.4. Descrierea algoritmilor fundamentali de prelucrare a grafurilor și implementarea acestora într-un limbaj de programare 1.5. Descrierea operațiilor specifice listelor simplu înlănțuite și elaborarea unor subprograme care să implementeze aceste operații 1.6. Descrierea operațiilor specifice structurilor arborescente și elaborarea unor subprograme care să implementeze aceste operații 1.7. Analizarea în mod comparativ a avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme	Structuri de date alocate dinamic* Grafuri neorientate și grafuri orientate <ul style="list-style-type: none"> • Terminologie • Tipuri speciale de grafuri • Reprezentarea grafurilor • Algoritmi de prelucrare a grafurilor - Parcurgerea grafurilor în lățime și în adâncime - Determinarea componentelor conexe ale unui graf neorientat - Determinarea componentelor tare conexe ale unui graf orientat - Determinarea matricei lanțurilor/drumurilor - Determinarea drumurilor de cost minim într-un graf - Arbori parțiali de cost minim Structuri de date arborescente <ul style="list-style-type: none"> • Arbori cu rădăcină • Arbori binari • Tipuri speciale de arbori binari - Arbore binar complet – definiție, proprietăți, reprezentare secvențială - Arbore binar de căutare – definiție, proprietăți, operații specifice - Heap-uri – definiție, proprietăți, operații specifice