

**EVALUARE: DIDACTICA INFORMATICII
2020 – 2021**

Evaluarea la cursul de Didactica Informaticii presupune ca fiecare student :

- să participe la realizarea unui proiect de grup axat pe proiectarea didactică a unei lecții de Informatică sau TIC;
- să prezinte un referat pe o temă la alegere;
- să rezolve o problemă și să prezinte demersul didactic necesar obținerii soluției de către elevi.

Temele de proiect de lecție / referat și problemele pot fi alese din listele de mai jos sau pot fi propuse de către studenți (însoțite de o scurtă descriere a relevanței acestora pentru curs).

**Data limită pt alegerea temelor (la curs/seminar sau prin email):
3 MARTIE 2021**

Coordonatorul grupului va trimite prin email un mesaj

- pe adresa smtataram@yahoo.com
- cu "subject"-ul: Alegere teme Didactica;
- care să indice:

- numărul și tema proiectului de grup (minim 3 optiuni)
- numărul și tema de referat aleasă de fiecare dintre membrii echipei (minim 3 optiuni pentru fiecare student)
- indicativul problemei alese de fiecare dintre membrii echipei (minim 3 optiuni pentru fiecare student)

Exemple:

G10... Consideratii metodice si software educational privind predarea notiunilor de teoria grafurilor neorientate

G22... Considerații metodice și software educațional privind predarea Informaticii la clasa a Va

G28... Metode de evaluare; software educational

FIRU Ion

R4 Personalități din domeniul Informaticii

R9 Evolutia sistemelor de operare

R11 Evolutia SGBD

P22

P2

P32

Sava Irina

R8 Limbaje naturale, limbaje formale, limbaje de programare¹

R13 Teorema de incompletețitudine Godel

R17 Algoritmi genetici

P62

P43

P44

Tătărâm Monica:

R5 Evoluția interfețelor om-calculator

R10 Tipuri speciale de baze de date

R31 Arhivare și restaurare

P1

P15

P10.

Mesajele care conțin opțiuni parțiale (numai referat și/sau problemă, numai proiect și/sau referat etc.) NU VOR FI LUATE ÎN CONSIDERARE și temele solicitate NU VOR FI REZERVATE.

Temele deja alese vor apărea în listele respective marcate cu roșu.

Discutarea și prezentarea preliminară a proiectelor, referatelor, problemelor:

3 martie - 7 aprilie 2021: ora 16-18,

¹ Ierarhia Chomsky, clasificări, paralele

NOTAREA PROIECTELOR / REFERATELOR / PROBLEMELOR

1. Fiecare proiect de grup / referat individual va avea alocate cate **80/20** min pentru prezentare și câte 10 min. pt discuții.

2. Prezentările vor fi făcute in ultimele săptămâni de școală, conform programării afișate pe grup la data de 24 martie.

3. Fișierele vor fi trimise prin email cu minimum o săptămână înainte de data prezentării de către conducătorul grupului. **Atenție: numele fișierelor trebuie să conțină numele și prenumele studentului, grupa, numărul de ordine al proiectului/referatului/problemei, titlul proiectului/referatului.**

Exemplu:

TataramMonica-231_G9-Algoritm.docx
TataramMonica-231_G9-Algoritm.pptx
TataramMonica-231_R5-EvolInterfete.docx
TataramMonica-231_R5-EvolInterfete.pptx
TataramMonica-231_P12.docx

Conditii de notare (orice tip de proiect, problema, orice tip de evaluare)

- nota la proiectul de grup: media aritmetică dintre nota grupului si nota individuală
- **se noteaza separat discutia manualelor** și, respectiv, proiectul didactic
- se noteaza separat realizarea tehnica și, respectiv, modul de prezentare a proiectului / referatului
- **se penalizează cu 1 punct fiecare saptamana de întârziere a trimerii fișierului respectiv** prin email precum și **nerespectarea convențiilor de identificare a fișierelor.**

• Nota finală:

- **Media aritmetică dintre nota pentru proiectul de grup, nota pentru referat, nota pentru problema rezolvată.**
- **Nota finală poate fi mărită prin punctele obținute pe parcursul semestrului ca urmare a răspunsurilor date la curs/seminar.**

Criterii de evaluare (proiect de grup)

2 puncte: concluzii formulate; argumentele aduse in sustinerea lor

2 puncte: modul de sustinere a prezentarii, colaborarea in timpul prezentarii, incadrarea în timp

1 puncte: realizarea prezentarii (conditii grafice, informatia continuta de ecran, comentariile si completarile facute, listing, fișier email)

2 puncte: bibliografie, originalitatea proiectului si a prezentarii; realizarea/utilizarea unui soft educational

1 puncte: diseminarea rezultatelor (cum le explica celorlalti ce au facut in proiect, cum raspund la intrebari, cum fac cunoscute celorlalti rezultatele proiectului)

2 puncte: intrebari/cpmentarii pe marginea prezentarii proiectelor colegilor

Criterii de evaluare (referat)

- 2 puncte:** numărul și volumul surselor consultate
- 2 puncte:** modul de susținere a prezentării, încadrarea în timp
- 2 puncte:** realizarea prezentării (captarea atenției, condiții grafice, informația conținută de ecran, comentariile și completările făcute, listing, fișier email)
- 2 puncte:** originalitatea proiectului și a prezentării
- 2 puncte:** diseminarea rezultatelor (formularea răspunsurilor la întrebări)

Criterii de evaluare (problemă)

- 2 puncte:** breviar teoretic în funcție de domeniul de cunoștințe (Programare C/Pascal; Metode de programare; Recursivitate; Complexitatea algoritmilor; Subprograme; Structuri de date (siruri de caractere, înregistrare, matrice); Teoria grafurilor)
- 3 puncte:** soluția completă, mai multe soluții diferite (cel puțin una într-un limbaj de programare ex. P11); evaluarea complexității; soluția optimă (ex. P24);
- 3 puncte:** tratarea metodică (formalizarea enunțului problemei practice; comentarea enunțului formalizat: complet? corect? dificil/banal?, ce confuzii pot apărea în enunț; exemplu și contraexemplu date de intrare; cum se poate «discretiza», pe ce probleme se bazează, cum se poate generaliza, cazuri particulare utile/ interesante; greșeli tipice în rezolvarea ei, formularea unei probleme similare și discutarea relației ei cu problema dată, în ce [punct al unui] proiect didactic poate fi utilizată;
- 2 puncte:** barem de corectare cu justificarea punctajului și propuneri pt soluții alternative.

Evident, codul trebuie să fie identat și să includă comentarii suficiente pentru înțelegerea algoritmului folosit.

Barem general (trebuie adaptat de student problemei alese):

Din oficiu	1 pct
Algoritm corect	5pct
Eficiență și analiză complexitate	2pct
Cunoștințe de limbaj	1pct
Explicații	1pct

TEME: REFERATE

Nr	Tema
R1	Primele calculatoare mecanice ²
R2	Primele calculatoare electronice ³
R3	Inventii recente în domeniul Informaticii (hard, soft, comunicare) ⁴
R4	Personalități din domeniul Informaticii ⁵
R5	Evoluția interfetelor om-calculator
R6	Algoritmi ⁶
R7	Evoluția structurilor de date ⁷ Structuri de date dinamice
R8	Limbaje naturale, limbaje formale, limbaje de programare ⁸
R9	Evoluția sistemelor de operare ⁹
R10	Tipuri speciale de baze de date ¹⁰
R11	Evoluția SGBD
R12	Teza Church-Turing ¹¹
R13	Teorema de incompletitudine Godel
R14	Calcul paralel și concurent ¹²
R15	Modele de calculabilitate neconventionale ¹³ DNA computing
R16	Inteligența artificială ¹⁴
R17	Algoritmi genetici
R18	Sisteme expert
R19	Matematica și Informatica (geometrie / algebra computation.)
R20	Științele exacte și Informatica
R21	Științele umaniste și Informatica
R22	Științele naturii și Informatica
R23	Arta și Informatica
R24	Rețele sociale
R25	Efectele dezvoltării Internet și web
R26	Cloud Computing
R27	Dihotomii în informatică ¹⁵

² Nepier, Schuckard, Pascal, Leibnitz, Babbage

³ ABC, Eniac,..., IBM 360/60; calculatoare românești

⁴ Disketa, softuri de criptare cu cheie publică, Sun, VisiCalc etc

⁵ John von Neumann, F.E. Codd, William Inmon, Tim Berners-Lee

⁶ Definiție (modele de calculabilitate), caracterizări, clasificare

⁷ Clasificări, tipuri speciale; http://en.wikipedia.org/wiki/List_of_data_structures

⁸ Ierarhia Chomsky, clasificări, paralele

⁹ Relația cu sistemele de calcul, Ms DOS, Windows, Unix, Linux, OS 2

¹⁰ DSS [3:694-729]; data mining [2:948-969]; data warehousing [2:912-948], [4:841-873]; temporale, spațiale, multimedia, distribuite, logice, GIS, digitale, WEB

¹¹ Clasică, modificată, teorii de complexitate, NP-completitudine

¹² Modele, legătura cu calculul clasic

¹³ Quantum computing, DNA computing, sisteme evolutive, P-sisteme

¹⁴ Concepte, aplicații

Nr	Tema
R28	Premiul Turing
R29	Virtualizare
R30	Centre de date
R31	Arhivare și restaurare
R32	Recuperarea datelor in caz de dezastru
R33	Securitatea datelor
R34	Utilizarea foilor de calcul excel pentru predarea altor discipline ¹⁶
R35	Realitate virtuala vs realitate augmentata
R36	Semantic Web

¹⁵ fiecare dohotomie: 5p bonificatie

Exemple: SO: ease of use vs. complete use of computing resources
software development & training: standardization vs. adaptability to new architecture, GUI
etc.

GUI: traditional visual aspect vs originality

security: security vs. ease of access to data / service

IS: in house development vs. outsourcing

¹⁶ Ex.: cu tabel in foaie și Insert function sau cu Goal Seek se pot evidentia relatiile dintre datele de intrare și datele de iesire (modif coef unui sistem linar il transforma din compat det in nedet sau incompatibil; modif coef. ec. gr. 2 transforma discriminantul din pozitiv in nul sau negativ și astfel și setul de solutii; modif coef. fctiei aduce pe grafic noi pcte de extrem sau de inflexiune sau modifica derivabilitatea sau continuitatea fctiei etc.)

TEME: PROIECTE DE GRUP (1 grup = 2-3 studenti)

Nr.	Tema
G1	Elemente de predare online
G2	Elemente de fixare a cunoștințelor online
G3	Elemente de evaluare a cunoștințelor online
G4	Consideratii metodice si software educational privind predarea Informaticii la clasa a Va
G5	Consideratii metodice si software educational privind predarea Informaticii la clasa a VIa
G6	Consideratii metodice si software educational privind predarea Informaticii la clasa a VIIa
G7	Consideratii metodice si software educational privind predarea Informaticii la clasa a VIIIa
G8	Consideratii metodice si software educational privind predarea notiunilor de structura a calculatorului
G9	Consideratii metodice si software educational privind predarea notiunii de algoritm
G10	Consideratii metodice si software educational privind predarea sistemelor de operare
G11	Consideratii metodice si software educational privind rezolvarea pe calculator a problemelor de aritmetica numerelor intregi
G12	Consideratii metodice si software educational privind predarea algoritmilor de sortare
G13	Consideratii metodice si software educational privind predarea structurilor de date statice
G14	Consideratii metodice si software educational privind predarea structurilor de date dinamice
G15	Consideratii metodice si software educational privind subprogramele
G16	Consideratii metodice si software educational privind reducerea recurentei la iteratie
G17	Consideratii metodice si software educational privind predarea notiunilor de teoria grafurilor neorientate
G18	Consideratii metodice si software educational privind predarea notiunilor de teoria grafurilor orientate
G19	Consideratii metodice si software educational privind predarea notiunilor de teoria arborilor oarecari
G20	Consideratii metodice si software educational privind predarea notiunilor de teoria arborilor binari
G21	Consideratii metodice si software educational privind predarea tehnicilor de programare
G23	Consideratii metodice si software educational privind predarea notiunilor de grafica
G24	Consideratii metodice si software educational privind predarea sistemelor de gestiune a bazelor de date

Nr.	Tema
G25	Consideratii metodice si software educational privind predarea notiunilor de birotica si multimedia
G26	Consideratii metodice si software educational privind rezolvarea pe calculator a problemelor de analiza numerica
G27	Consideratii metodice si software educational privind predarea principiilor programarii structurate
G28	Consideratii metodice si software educational privind predarea principiilor programarii orientate spre obiect
G29	Consideratii metodice si software educational privind predarea principiilor programarii vizuale
G30	Etapele realizarii unui proiect didactic; software educational
G31	Metode de captare a atentiei si motivare a elevilor la orele de informatica; software educational
G32	Metode moderne de predare-invatare; software educational
G33	Metode de evaluare; software educational
G34	Metode moderne in procesul de comunicare elev – profesor; software educational
G35	Proiectarea unui curriculum de Informatică la nivel universitar
G36	Proiectarea unui curriculum de Informatică la nivel preuniversitar

LISTA DE PROBLEME

1. Fie A un șir de n numere, $n \geq 3$.

Să se scrie un program care să înlocuiască fiecare element nül din șir, a_k , $2 \leq k \leq n-1$, cu:

- a) media geometrică a celui mai mare și a celui mai mic element din șir;
- b) media geometrică a modulului vecinilor săi direcți.

2. Fie numărul natural $n \geq 10$.

Să se elaboreze un program care să determine și să afișeze cea mai mare și cea mai mica cifră a numărului precum și câtuł acestora.

3. Se consideră două numere naturale strict pozitive a și b reprezentabile în calculator. Să se scrie proceduri/funcții pentru listarea:

- a) numărului c obținut prin inversarea ordinii cifrelor lui b ;
- b) celui mai mic multiplu comun al lui a și c .

4. Fie numărul natural $n \geq 10$.

a) Să se elaboreze un program care să determine dacă numărul este palindrom; în caz afirmativ se cere să se tipărească numărul și un mesaj (de exemplu: Numarul 12421 este palindrom); în caz negativ se cere să se afișeze numărul și oglinditul său (de exemplu: Numarul 186335 este oglinditul numarului 533681).

b) Să se determine și să se afișeze numărul pozițiilor în care numărul n și oglinditul său coincid (adica: pe poziția respectivă în cele două numere apare aceeași cifră).

5. Se citesc $n \geq 3$ numere de la tastatură. Să se elaboreze un program care să determine și să afișeze cele mai mari 3 numere dintre acestea.

6. Să se ordoneze crescător 3 numere naturale, citite de la tastatură.

7. Un numar de 500 de candidati concureaza la un examen de admitere pentru 250 de locuri. Notele obtinute de ei sunt cuprinse între 1 și 10 si sunt memorate într-un vector V . Sa se scrie proceduri sau functii pentru:

- a) Ordonarea descrescatoare a punctajelor obtinute la examen.
- b) Determinarea punctajul ultimului candidat admis, avand in vedere ca nota minima de admitere este 6.

8. Se dau $n-1$ numere între 1 și n , $n \in \mathbb{N}$; sa se determina intregul care lipseste. Sa se propuna si algoritmul de complexitate minima

9. Se citește o secvență de n numere naturale a_1, a_2, \dots, a_n . Să se sorteze această secvență în ordine crescătoare fără a face nicio comparație între elementele ei.

10. Fie $n \in \mathbb{N}$, $n \geq 2$.

- a) Sa se verifice daca n este prim.
- b) Sa se afișeze descompunerea in factori primi a lui n .

Exemple:

$n = 7$

7 este prim

$7 = 7$

$n = 1960$

1960 nu este prim

$1960 = 2^3 \cdot 5 \cdot 7^2$

11. Să se elaboreze un program care să determine și să afișeze cmmdc și cmmmc a două numere naturale citite de la tastatură.

12. Să se elaboreze un program care să determine și să afișeze toți divizorii primi ai unui număr natural n , citit de la tastatură.

13. Să se elaboreze un program care să determine și să afișeze toate tripletele de numere naturale a, b, c cu proprietățile:

(i) $1 < a < b < c < 100$

(ii) $(a + b + c)$ se divide cu 10

14. Să se elaboreze un program care să determine și să afișeze toate tripletele de numere naturale a, b, c cu proprietățile:

(i) $1 < a < b < c < 100$

(ii) $a^2 + b^2 = c^2$

15. Se dă secvența de numere 1, 3, 9, 27, 81, și un număr $k > 20$.

Să se elaboreze un program care să determine și să afișeze al k -lea număr din secvență.

16. Se citesc de la tastatură m , respectiv n cifre; fiecare dintre acestea este cuprinsă între 0 și $b-1$, unde b este un număr natural, $2 \leq b \leq 9$. Cu primele m cifre se formează un număr natural M iar cu celelalte n cifre se formează un alt număr natural N (se presupune că numerele naturale m și n sunt suficient de mari pentru ca atât M cât și N să nu poată fi reprezentabile în memorie).

Să se elaboreze un program care să determine și să afișeze:

a) cel mai mare dintre aceste numere;

b) suma lor.

17. Fie un vector V cu n ($n \geq 2$) elemente numere întregi.

a) Să se construiască și să se afișeze vectorul W care conține elementele lui V în ordine inversă.

b) Să se determine și să se afișeze numărul pozițiilor în care cei doi vectori V și W coincid (adică: pe poziția respectivă cei doi vectori au elemente egale).

17bis). Fie V un vector cu n elemente numere întregi ($n > 2$). Să se determine cel mai mic număr care se poate forma prin juxtapunerea cifrei minime a fiecarei componente din V . (Exemplu: $n=6$, $V=(12, 21, 333, 404, 5206, 2) \Rightarrow 1,2,3,0,2 \Rightarrow 100223$)

18. Să se determine simultan, prin metoda Divide et Impera maximul și minimul dintr-un tablou de n numere întregi.

19. Sa se verifice - iterativ și apoi recursiv - egalitatea a doua siruri de caractere introduse de la tastatura.

20. Se citesc un numar natural n si n numere intregi. Sa se determine si sa se afiseze o submultime a celor n numere a caror suma se divide cu n . Se cere un algoritm cat mai eficient.

21. Fie un șir de n numere întregi și un numar întreg x . Să se scrie un subprogram recursiv sau iterativ care să partiționeze șirul în două subșiruri astfel încât unul să conțină elementele mai mici decât x iar celalalt elementele mai mari sau egale cu x . Observație: ordinea elementelor nu este semnificativă.

22. Sa se scrie un program care citeste de la tastatura 3 siruri de caractere a, b si c . Sirurile a si b au aceeași lungime si contin fiecare doar caractere diferite. Sa se afiseze un sir d obtinut din sirul c , copiind toate caracterele si inlocuind orice caracter din sirul a cu caracterul corespunzator din sirul b .

23. Se citesc de la tastatura numarul natural $n \geq 2$ si sirul de numere naturale a_1, a_2, \dots, a_n . Sa se scrie un program (eventual cu complexitate timp liniară) care să afișeze indicii i și j care indeplinesc simultan urmatoarele conditii:

a) $1 \leq i < j \leq n$;

b) $a_i > a_k$ si $a_j > a_k$, pentru orice k , $i+1 \leq k \leq j-1$;

c) diferenta $j-i$ este maxima.

24. Se numeste subsecvență a unui vector V cu n elemente numere intregi un vector cu cel puțin un element si cel mult n elemente care se gasesc pe pozitii consecutive in vectorul V . Se cere să se scrie un program (eventual cu complexitate timp liniară) care să citească de la tastatura numarul natural n si vectorul V avand n elemente intregi si să afișeze cea subsecventa a lui V care are suma elementelor maximă.

25. Sa se scrie un program care sa citeasca doua numere naturale $m, n \geq 1$ si doua siruri de numere naturale $s_1 = (a_1, a_2, \dots, a_n)$ si $s_2 = (b_1, b_2, \dots, b_m)$ și care să afișeze:

0 daca $s_1 = s_2$;

1 daca $s_1 > s_2$;

-1 daca $s_1 < s_2$.

Explicatie:

Ordinea lexicografica pe multimea sirurilor de numere naturale se defineste astfel: $s_1 > s_2$ \Leftrightarrow

a) Exista i , $1 \leq i \leq \min(m, n)$, cu proprietatea $a_j = b_j$ pentru orice $1 \leq j < i$ si $a_i > b_i$ sau

b) $n > m$ si $a_j = b_j$ pentru orice $1 \leq j \leq m$.

26. Fie doua siruri de numere intregi pozitive $(x_n)_{n \in \mathbb{N}}$ si $(y_n)_{n \in \mathbb{N}}$, definite prin:

$$x_0=1, y_0=0,$$

$$x_{n+1}=ax_n+dy_n,$$

$$y_{n+1}=x_n+ay_n.$$

unde $a \in \mathbb{N}$, $a > 1$, $d=a^2-1$. Sa se scrie un program care sa determine si sa afiseze, pentru numerele naturale nenule a si k citite de la tastatura, cea de-a k -a pereche (x_k, y_k) .

27. Se citeste de la tastatura sub forma de sir de caractere o expresie formata numai din paranteze rotunde, precum si un numar natural nenul n . Sa se găsească un algoritm liniar dupa numărul de caractere din sirul citit care sa verifice dacă expresia este corect parantezata și are nivelul de imbricare cel mult n .

Exemple.

$n=2$: expresia $((()))()$ indeplineste ambele cerințe; expresia $((()))()$ nu este corect parantezata, deci nu indeplineste prima cerința; expresia $((())())()$ are nivelul de imbricare 3 (mai mare ca 2), deci nu indeplineste a doua cerinta.

28. Scrieti un program care citeste o expresie ce poate utiliza paranteze rotunde, drepte si acolade si verifica daca are parantezele puse corect; se ignora restul componentelor expresiei.

Exemple:

$x(aa[bcd]a(\{1+2\}x))$: Corect

$x(aa[bcd]a(\{1+2\}x))$: Gresit

29. (Master 2010) Se citeste de la tastatură un polinom P de grad n sub forma unui vector de coeficienti reali, precum si un numar real a . Sa se elaboreze un algoritm liniar care sa calculeze și să afișeze valoarea $P(a)$.

30. Fie $n, m \in \mathbb{N}$, $n > m \geq 0$. Sa se calculeze C_n^m , unde se notează cu

$$C_n^m = n! / (m!(n-m)!)$$

31. Intr-o sală trebuie programate N spectacole. Pentru fiecare spectacol se cunoaște intervalul $[s, f)$ de desfășurare. Să se determine numărul maxim de spectacole ce pot fi programate astfel încât ele să nu se suprapună.

Exemplu:

pentru $n=8$ și $[10,15)$ $[2,4)$ $[7,9)$ $[21,25)$ $[10,12)$ $[12,15)$ $[7,8)$ $[20,27)$

se va afișa: Spectacolele alese sunt 2 7 5 6 4

Numărul de spectacole este 5

32. Fie un grup G de n persoane, $n > 2$, care se cunosc sau nu intre ele, adica: persoana x cunoaste persoana y , dar persoana y nu trebuie neaparat sa cunoasca persoana x (aceasta relate de „cunoștință” nu este intotdeauna simetrica). Pentru ca un membru al grupului G sa fie considerat „o celebritate” trebuie ca el sa fie cunoscut de toti ceilalti membri ai grupului, dar el trebuie să nu cunoasca pe nici un alt membru al grupului. Sa se scrie un algoritm care sa determine daca in grupul dat exista sau nu cel puțin o celebritate și in caz afirmativ sa afiseze toate persoanele care sunt celebritati.

33. Fie A o matrice pătratică cu elemente numere naturale. Să se scrie un program care să afișeze, pentru fiecare linie: cel mai mare element, cel mai mic element și un mesaj care să indice dacă acestea formează o pereche de numere prime între ele.

34. Fie A o matrice pătratică de numere întregi, de ordin n, $n \geq 2$. Să se scrie un program care să afișeze cel mai mare divizor comun și cel mai mic multiplu comun al elementelor de pe diagonala principală.

35. Fie o matrice pătratică A de dimensiune $n \geq 2$ peste mulțimea numerelor întregi. Să se elaboreze un program pentru afișarea unui mesaj care să indice dacă cel mai mare și cel mai mic element din A formează o pereche de numere prime între ele.

36. Fie o matrice A de dimensiuni n, $m \geq 2$ cu elemente numere întregi. Să se scrie un program care să afișeze elementele a_{ij} (un element $a_{ij} \in A$, $1 \leq i \leq n$, $1 \leq j \leq m$, se numește a_{ij} dacă el reprezintă simultan minimul dintre elementele de pe linia i și maximul dintre elementele de pe coloana j).

37. Fie o matrice pătratică A de dimensiune $n \geq 2$ peste mulțimea numerelor întregi. Să se atribuiască unei variabile logice P valoarea TRUE dacă este îndeplinită cel puțin una dintre următoarele 2 condiții:

(a) toate elementele de pe diagonala principală sunt numere prime;

(b) $\sum_{j=1}^n A_{i,j} \leq A_{i,i} \leq \sum_{j=1}^n A_{j,i}, \forall 1 \leq i \leq n$

38. Fie o matrice A cu n linii și m coloane, cu elemente numere reale ($n, m \geq 2$). Se dau numerele $1 \leq k_1, k_2, k_3, k_4 \leq \max\{n, m\}$ și $k_1 \leq k_3, k_2 \leq k_4$. Să se elaboreze un program care să calculeze și să afișeze media aritmetică a elementele subdreptunghiului cu colțul stanga sus de coordonate (k_1, k_2) și colțul din dreapta jos de coordonate (k_3, k_4) . Exemplu:

$$A = \begin{pmatrix} 1,1 & 5,5 & 3,3 & 4,2 & 5 \\ 0,6 & 7,3 & 8 & 9 & 10 \\ 5,5 & 3,3 & 4,2 & 7,3 & 8 \\ -1,2 & -1,3 & 1,2 & 6 & 14 & 15 \end{pmatrix}, k_1 = 1, k_2 = 2, k_3 = 2, k_4 = 4$$

Rezultat: $\frac{5,5 + 3,3 + 4,2 + 7,3 + 8 + 9}{6} = 11,17$

39. Fie o matrice A cu n linii și m coloane, cu elemente numere reale ($n, m \geq 2$). În A sunt permise doar două operații: permutarea a 2 linii între ele, respectiv a două coloane între ele. Să se transforme matricea în așa fel încât elementele de pe diagonala principală să fie ordonate descrescător.

40. Fie $\rho \subseteq A \times A$ și $\rho' : A \times A \rightarrow A$ unde $\text{card}(A) = n$, $10 \leq n \leq 100$:

Să se elaboreze un program care să verifice dacă

a) ρ este o relație de echivalență pe A;

b) (A, ρ') este grup comutativ.

41. Fie mulțimea finită $A = \{1, 2, \dots, n\}$ și fie L o lege de compoziție definită pe M. Să se scrie proceduri/funcții care să verifice dacă:

a) L este lege de compoziție internă pe A.

b) Legea L este comutativă.

c) (A, L) este grup abelian.

42. Se da o secvență de întregi cititi pe rand de la tastatura.
- Să se insereze fiecare întreg citit într-o structură de date.
 - Să se parcurgă structura cu afișarea întregilor în ordine crescătoare.
43. Fie G un digraf, aciclic în care orice drum are maximum $k \geq 2$ arce.
Să se elaboreze un program care să determine și să afișeze o partiție a mulțimii vârfurilor lui G în cel mult k părți, astfel încât pentru orice două noduri x și y din aceeași submulțime să nu existe drum nici de la x la y nici de la y la x .
44. Fie G un digraf cu $n \geq 2$ noduri. Să se atribuie unei variabile logice P valoarea TRUE dacă este există în G un nod având gradul interior $n-1$ și gradul exterior 0.
45. Se dau n puncte în plan. Să se elaboreze un program care să determine și să afișeze distanța minimă între toate perechile de puncte date.
46. Se dau n ($n \leq 10000$) segmente pe o dreaptă, segmente colorate în culori diferite. Să se calculeze lungimea porțiunii colorate din dreaptă. Citirea și afișarea datelor se va face folosind fișiere, conform exemplului.
- Exemplu:*
- | SEGM.IN | SEGM.OUT |
|---------|----------|
| 5 10 | 15.20 |
| 6 12 | |
| 20 28.2 | |
47. Se dau n ($n \in \mathbb{N}$, $2 \leq n \leq 10.000$) segmente disjuncte pe o dreaptă, specificate prin extremități și colorate cu p ($p \in \mathbb{N}$, $2 \leq p \leq 100$) culori, fiecare segment fiind colorat cu o singură culoare. Să se elaboreze un program care să determine și să afișeze:
- sumele lungimilor segmentelor colorate cu fiecare dintre cele p culori folosite;
 - culoarea pentru care aceasta sumă este maximă.
48. Se dau n dreptunghiuri ($n < 101$) cu laturile paralele cu axele prin coordonatele lor carteziane, numere întregi mai mici decât 2^{15} citite de la tastatură. Să se afișeze aria totală acoperită de cele n dreptunghiuri.
49. Să se scrie un program cu 4 argumente, primul reprezentând numele unui fișier text de intrare, iar al doilea numele unui fișier text de ieșire. Argumentele 3 și 4 au aceeași lungime și nu conțin același caracter de mai multe ori. Să se construiască fișierul de ieșire pornind de la fișierul de intrare, copiind toate caracterele și înlocuind orice caracter din argumentul 3 cu caracterul corespunzător din argumentul 4. Se va emite mesaj de eroare în situația în care numărul de argumente este incorect sau argumentele 3 și 4 nu au aceeași lungime sau argumentele 3 și 4 conțin vreun caracter de mai multe ori.
50. Să se scrie proceduri recursive și iterative pentru funcția lui Ackermann-Peter, funcția Manna-Pnueli, sirul lui Fibonacci, calcularea cmmdc (la alegere).

$$\text{AckP}(m,n) = \begin{cases} n+1, & m=0 \\ \text{AckP}(m-1,1), & n=0 \\ \text{AckP}(m-1, \text{AckP}(m,n-1)), & \text{altfel} \end{cases}$$

$$\text{MannaPn}(x) = \begin{cases} x-1, & x \geq 12 \\ \text{MannaPn}(\text{MannaPn}(x+2)), & x < 12 \end{cases}$$

51. Se numește număr “bine ordonat crescător” un număr natural cu proprietatea că cifrele sale citite de la stânga la dreapta sunt în ordine crescătoare.

Exemplu: 3478.

Fiind dat un număr natural citit de la tastatură, să se stabilească dacă este “bine ordonat crescător” sau nu, afișându-se un mesaj corespunzător.

52. Se numește număr “bine ordonat descrescător” un număr natural, cu proprietatea că cifrele sale citite de la stânga la dreapta sunt în ordine descrescătoare.

Exemplu: 9653.

Să se afișeze toate numerele “bine ordonate descrescător” cu trei cifre.

53. Se citește de la tastatură un număr n natural par. Să se descompună în sumă de două numere prime (conjectura lui Goldbach).

54. Un număr natural se numește perfect dacă este egal cu suma divizorilor săi, fără el însuși.

Exemplu: $6=1+2+3$.

Să se verifice dacă un număr natural dat este perfect.

55. Să se afișeze toate numerele perfecte situate în intervalul $[p, q]$ și numărul acestora (p și q date).

56. Se citesc de la tastatură n numere naturale. Să se afișeze numerele care au cea mai mare sumă a divizorilor proprii.

57. Se numește număr “rotund” un număr natural cu proprietatea că reprezentarea sa binară are același număr de cifre de 0 și de 1.

Exemplu: $37_{10} = 100101_2$. **Contraexemplu:** $5_{10} = 101_2$

Să se verifice dacă un număr natural dat este “rotund”.

58. Să se afișeze toate numerele rotunde situate în intervalul $[p, q]$ și numărul acestora (p și q date).

59. Se citesc două numere întregi. Să se afișeze în câte zerouri se termină produsul lor, fără a calcula produsul.

60. Se numește număr “prim în sens tare” un număr natural cu proprietatea că atât el cât și răsturnatul său sunt prime.

Exemplu: 2, 3, 5, 7, 11, 13, 17. **Contraexemplu:** 23, 53

Să se verifice dacă un număr natural dat este "prim în sens tare".

61. Să se afișeze toate numerele mai mici sau egale cu un număr natural **n** citit de la tastatură, care au proprietatea că sunt "prime în sens tare".

62. Scrieți apoi un program care: i) citește un număr natural **n** format din cel mult 4 cifre și două cifre nenule **c1** și **c2**; ii) modifică numărul **n** prin înlocuirea fiecărei apariții a cifrei **c1** cu cifra **c2**; iii) afișează numărul astfel obținut.

Transformați programul de mai sus într-un subprogram (adică: scrieți definiția completă a subprogramului **numar**, cu trei parametri, care primește prin intermediul parametrului **n** un număr natural format din cel mult 4 cifre, iar prin intermediul parametrilor **c1** și **c2** câte o cifră nenulă. Subprogramul caută fiecare apariție a cifrei **c1** în **n** și, dacă aceasta apare, o înlocuiește cu **c2**. Subprogramul returnează tot prin intermediul parametrului **n** numărul astfel obținut. Dacă cifra **c1** nu apare în **n**, atunci valoarea lui **n** rămâne nemodificată.).