



Simulare pentru EXAMENUL DE BACALAUREAT - decembrie 2025

Proba E.d)
INFORMATICĂ

Varianta 1

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică **Filieră vocațională, profil militar, specializare matematică-informatică**

- Toate subiectele sunt obligatorii.
- Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

- Variabila **a** este de tip real. Pentru a verifica dacă valoarea variabilei $a \in [-5, 2] \cup \{3, 5, 8\}$ se va utiliza următoarea expresie C++:
 a. `!(a<-5) || (a>2) || (a==3) || (a==5) || (a==8)`
 b. `(a>=-5) && (a<=2) && (a==3) || (a==5) || (a==8)`
 c. `(a>-5) && (a<-5) || (a>2) && (a==3) && (a==5) && (a==8)`
 d. `(a>=-5) || !(a>2) || (a==3) || (a==5) || (a==8)`
- Subprogramul **f** este definit alăturat. Ce se afișează în urma apelului **f(2)** ?
 a. 3 2 0 1 2
 b. 0 2 1 3 2
 c. 0 2 2 1 3
 d. 2 1 3 2

```
void f(int x)
{ if (x>0)
  { f(x-1);
    cout<< x+1 <<" ";
  }
  cout<<x<<" ";
}
```
- Un sistem informatic trebuie să aloce resurse (numere naturale nenule) către mai multe procese, astfel încât suma totală a resurselor alocate să atingă o valoare fixă. Se generează, prin metoda backtracking, toate distribuțiile distincte de resurse pentru care suma elementelor este egală cu 6, obținându-se următoarele combinații: {1,2,3}, {1,5}, {2,4}, {6}. Folosind aceeași metodă, se generează distribuții distincte de resurse a căror sumă este egală cu 10. Stabiliți în ce ordine sunt generate următoarele distribuții: 1) {2, 3, 5}; 2) {3, 7}; 3) {2, 8}; 4) {1, 9}.
 a. 4 1 2 3 b. 4 2 1 3 c. 4 1 3 2 d. 4 2 3 1
- Precizați care este numărul muchiilor care trebuie adăugate, într-un graf neorientat conex care are n noduri și $n-1$ muchii (n număr natural cunoscut), pentru a deveni complet:
 a. $n \cdot (n-1) / 2$ b. 0 c. $(n+1) \cdot n / 2$ d. $(n-1) \cdot (n-2) / 2$
- Se consideră un graf orientat cu 6 noduri, numerotate 1,2,...,6. Arcele grafului sunt de forma $(x, 2 \cdot x)$ pentru orice $x \in \{1, 2, 3\}$ și de forma $(x, x-1)$ pentru orice $x \in \{2, 3, 4, 5, 6\}$. Care este numărul minim de arce ce trebuie adăugate astfel încât graful să fie tare conex?
 a. 3 b. 2 c. 0 d. 5

SUBIECTUL al II-lea

(40 de puncte)

- Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $x \times y$ restul împărțirii numărului natural x la numărul natural nenul y .

- Scrieți valoarea care se va afișa la finalul executării algoritmului dacă se citesc valorile $x=15$ și $y=25$. (6p.)
- Dacă pentru y se citește valoarea 3, scrieți toate cifrele care pot fi citite pentru variabila x astfel încât, pentru fiecare dintre acestea, la finalul executării algoritmului să se afișeze o valoare de forma 2^p , unde p este un număr natural. (6p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
citește x,y
      (numere naturale)
cât timp y>0 execută
  cât timp x>=y execută
    x←x-y
  z←x
  x←2*y
  y←2*z
scrie x
```

- d. Scrieți în pseudocod un algoritm echivalent cu cel dat în care prima structură `cât timp...execută` să fie înlocuită cu o structură repetitivă cu test final. (6p.)
2. Se definește tipul de date **punct**, utilizat pentru memorarea coordonatelor (abscisa și ordonata) unui punct din plan, numere reale. Se consideră un punct P în plan. De asemenea, se definește tipul de date **segment**, care memorează două puncte distincte A și B , reprezentând extremitățile unui segment din plan.
- ```
struct punct struct segment
{ float x,y; }P; { punct A,B; }S;
```
- Scrieți o expresie în limbajul C/C++ care să aibă valoarea 1 dacă și numai dacă punctul  $P$  are aceleași coordonate cu mijlocul segmentului  $[AB]$ . (6p.)
3. Variabilele  $i, j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 4 linii și 5 coloane, numerotate de la 0, cu elemente numere întregi, inițial toate nule. Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni astfel încât, în urma executării acesteia, variabila  $a$  să memoreze tabloul alăturat. Se vor utiliza maxim trei instrucțiuni de atribuire. (6p.)

|    |    |    |    |    |
|----|----|----|----|----|
| 5  | 4  | 3  | 2  | 1  |
| 6  | 7  | 8  | 9  | 10 |
| 15 | 14 | 13 | 12 | 11 |
| 16 | 17 | 18 | 19 | 20 |

**SUBIECTUL al III-lea****(30 de puncte)**

1. Un număr natural  $n$  se numește aproape prim dacă se scrie ca produs de două numere naturale prime distincte, de exemplu  $6=2*3$  este aproape prim, iar  $12=2*2*3$  nu este aproape prim. Scrieți definiția completă a unui subprogram **AproapePrime** cu patru parametri:
- $a$  și  $b$  – prin care primește două numere naturale ( $1 \leq a \leq b \leq 109$ )
  - $c$  și  $d$  – prin care furnizează cel mai mic și cel mai mare număr aproape prim din intervalul  $[a, b]$  sau 0 (ambii parametri), dacă intervalul nu conține cel puțin două astfel de valori.

**Exemplu:** pentru  $a=5$  și  $b=16$  atunci după apel  $c$  va fi 6 și  $d$  va fi 15.

**(10p.)**

2. Două șiruri de caractere se numesc *anagrame* dacă au aceleași caractere, eventual dispuse în altă ordine. Se citește de la tastatură două șiruri de caractere: un șir  $s$  cu cel mult 100 de caractere litere mici și spații și un cuvânt  $c$  cu cel mult 20 de caractere litere mici. Scrieți un program C/C++ prin care să se construiască în memorie și să afișeze pe ecran un șir de caractere care să conțină, separate printr-un spațiu, toate cuvintele din șirul  $s$  care nu sunt anagramele cuvântului  $c$ , sau mesajul "nu există" dacă șirul  $s$  nu conține astfel de cuvinte.

**Exemplu 1:**

dacă pentru variabila  $s$  a fost citit șirul: *anagrama numelui adina este diana scris cu litere mici*

iar șirul  $c$  este: *diana*

atunci șirul construit va fi: *anagrama numelui este scris cu litere mici*

**Exemplu 2:**

pentru șirul  $s$ : *adina este numele ei*

și cuvântul  $c$ : *dana*

se va afișa mesajul "nu există".

**(10p.)**

3. În fișierul **bac.in** se află pe prima linie un număr natural  $c$  și pe a doua linie se află cel mult 106 numere naturale separate printr-un spațiu, toate valorile din fișier fiind numere din intervalul  $[0, 9]$ . Să se scrie un program C/C++ care citește valorile din fișier și afișează pe ecran lungimea maximă a unei secvențe de valori cu proprietatea că cifra  $c$  apare în secvență de exact două ori. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

**Exemplu:** dacă fișierul conține valorile:

5

1 2 3 1 5 0 1 4 8 5 9 9 3 2 5 8 9 2 6 7 3 5 6 7 8

Pe ecran se va afișa 16.

- a. Descrieți algoritmul de rezolvare al problemei enunțate precizând elementele de eficiență. (2p.)

- b. Scrieți programul C/C++ corespunzător algoritmului descris. (8p.)