

SIMULARE EXAMENUL DE BACALAUREAT – decembrie 2025

Proba E. d)

Proba scrisă la INFORMATICĂ

Limbajul C/C++

Simulare

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei 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.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

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.

1. Precizați ce valoare va avea variabila reală **x**, după executarea următoarei instrucțiuni **C/C++**.
x = 7.51+35/4*67%8-2.83;
a) Instrucțiunea este incorectă
b) 4
c) 4.68
d) 28.68
 2. Precizați care este numărul muchiilor care trebuie adăugate într-un graf neorientat cu **n** noduri și **n-1** muchii, pentru a deveni complet:
a) $(n-1) * (n-2) / 2$
b) $n * (n-1) / 2$
c) $(n+1) * n / 2$
d) n
 3. Folosind metoda backtracking se generează toate șirurile formate din patru caractere distincte din mulțimea {#, *, &, @, %}. Primele trei soluții sunt: #* & @, #* & %, #* @ &. Indicați care este soluția generată înainte de &*# @ .
a) * @ % &
b) * & % @
c) & # @ %
d) & # % @
 4. Precizați care din următoarele șiruri de numere poate reprezenta gradele vârfurilor unui arbore cu 6 noduri.
a) 1 1 1 2 1 3
b) 1 1 1 2 1 4
c) 1 1 1 2 1 5
d) 1 1 1 2 1 6
 5. Subprogramul **f** este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.

```
cout << f(16);
```

```
int f (int x)
{ if( x > 8) return f(f(x-3)) + 4;
  else return x - 5;}
```
- a) -5 b) -2 c) -1 d) 1

SUBIECTUL II

(40 de puncte)

1. **Algoritmul alăturat este reprezentat în pseudocod.**
S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[c]$ partea întreagă a numărului real c .
- a. Scrieți numărul afișat în urma executării algoritmului dacă pentru a se citește valoarea 5 iar pentru n valoarea 4. **(6p.)**
- b. Scrieți două seturi de date distincte care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea 64. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura repetitivă cât timp... execută cu o structură repetitivă de alt tip. **(6p.)**

```

citește a, n (numere naturale
nenule)
p ← 1
cât timp n ≠ 0 execută
    c ← n % 2
    dacă c=1 atunci
        p ← p * a
    n ← [n / 2]
    a ← a * a
scrie p

```

2. Fiecare dintre variabilele A și B, declarate mai jos, memorează coordonatele întregi (x abscisa, iar y ordonata) ale câte unui punct în sistemul de coordonate xOy , extremități ale unui segment.

```
struct Punct
{int x, y;} A,B;
```

Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă segmentul cu extremitățile în punctele corespunzătoare variabilelor A și B intersectează axa Ox a sistemului de coordonate. **(6p.)**

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 5 linii și 5 coloane având inițial toate elementele nule. Fără a utiliza alte variabile decât cele menționate, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul de mai jos.

```
1 1 1 1 1
1 2 3 4 5
1 3 6 10 15
1 4 10 20 35
1 5 15 35 70
```

```
for(i = 0; i < 5; i++)
    for(j = 0; j < 5; j++)
        .....
(6p.)
```

SUBIECTUL III

(30 de puncte)

1. Subprogramul **cifre** primește prin intermediul parametrului **n** un număr natural de minimum 2 și maximum 9 cifre care conține cel puțin o cifră pară și cel puțin o cifră impară și furnizează prin intermediul parametrului **nrmin** cel mai mic număr care se poate forma cu cifrele pare ale lui n, iar prin intermediul parametrului **nrmax** cel mai mare număr care se poate forma cu cifrele impare ale lui n.

Exemplu: dacă **n=276025**, atunci, după apel, **nrmin=2026**, iar **nrmax=75**.

(10p.)

2. Alexandra a învățat la școală despre șiruri de caractere. Profesorul le-a dat următoarea problemă ca temă:

Se consideră un șir de cel mult 100 de caractere format din litere mari și mici ale alfabetului englez separate prin spații. Din șir trebuie eliminate cuvintele care conțin cel puțin o literă mare.

Scrieți un program C/C++ care citește de la tastatură un astfel de text și construiește în memorie și apoi afișează pe ecran șirul format din cuvintele ce conțin doar litere mici, cuvintele fiind separate prin exact un spațiu. Dacă nu există cuvinte formate doar din litere mici, atunci se va afișa mesajul **nu exista**.

Exemplu: dacă se citește textul **Ana are DOAR mere**, atunci se va afișa **are mere**.

(10p.)

3. Se consideră o banchiză pe care sunt așezați, în linie, mai mulți pinguini. Se cunosc înălțimile acestora, exprimate prin numere naturale. Se știe că pinguinii *Imperiali* sunt cei a căror înălțime este cuprinsă între două valori date **a** și **b** (adică valori aflate în intervalul $[a, b]$).

Se cere să se determine ultima secvență de lungime maximă formată numai din pinguini *Imperiali*. O secvență este un șir de cel puțin 2 elemente din linie aflate pe poziții consecutive.

Fișierul **bac.txt** conține maximum 10^5 numere naturale de cel mult 4 cifre fiecare: pe prima linie valorile **a** și **b** iar pe următoarea linie un șir de numere naturale reprezentând înălțimile pinguinilor de pe banchiză. Se vor afișa pe ecran, separate printr-un spațiu, două valori **st** și **dr** reprezentând capetele ultimei secvențe dacă aceasta există, sau valoarea **-1** în caz contrar.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu1: dacă fișierul conține valorile alăturate, atunci pe ecran se afișează 13 15

bac.txt

```
7 10
1 2 7 8 10 4 5 8 8 9 11 12 9 9 10 12
```

Exemplu2: dacă fișierul conține valorile alăturate, atunci pe ecran se afișează -1

bac.txt

```
7 10
1 2 3 4 5 6 7 11 8
```

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

b) Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**