

Examenul național de bacalaureat 2026
Proba E. d)
INFORMATICĂ
Limbajul C/C++

Simulare

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-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. Variabila x este de tip întreg. Indicați o expresie care are valoarea 1 dacă $x \leq 20$ || $x > 26$ și numai dacă expresia C/C++ alăturată are valoarea 1.
- a. `!(x>20) || !(x<=26)` b. `!(x>=20) && !(x>=26)`
c. `!(x<20 || x<=26)` d. `!(x<20 && x<26)`
2. Un mozaic este alcătuit din plăcuțe de formă pătrată, cu dimensiuni egale, de culoare albă sau roșie, așezate pe două rânduri și aliniată ca în exemplu. Mozaicul are un model în zig-zag dacă oricare două plăcuțe cu o latură comună au culoare diferite. Se știe că plăcuțele de pe latura din dreapta au culori diferite. Variabilele j și ok sunt întregi, iar variabila a este un tablou bidimensional cu două linii și 12 coloane, cu elemente de tip `char`, în care fiecare linie, în ordine, memorează succesiunea de plăcuțe aflate pe un rând al mozaicului, notându-se cu litera **R** cele roșii și cu litera **A** cele albe. Liniile și coloanele sunt numerotate începând de la 0. Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței C/C++ obținute, variabila ok să aibă valoarea 1, dacă mozaicul are un model în zig-zag, sau valoarea 0 în caz contrar. **Exemplu:** în urma executării secvenței pentru tabloul de mai jos, unde sunt evidențiate două zone în care nu se respectă modelul, $ok=0$.
- ```
ok=1;
for(j=0;j<11;j++)
 if(.....) ok=0;
```
- |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R | A | A | R | A | R | A | R | A | R | A | R |
| A | R | R | A | R | A | R | A | R | R | R | A |
- a. `a[0][j]==a[0][j-1] || a[0][j]==a[1][j-1]`                      b. `a[0][j]==a[0][j+1] || a[0][j]==a[1][j+1]`  
c. `a[0][j]==a[0][j-1] || a[0][j]==a[1][j+1]`                      d. `a[0][j]==a[0][j+1] || a[0][j]==a[1][j]`
3. Subprogramul  $f$  este definit alăturat, iar variabila  $x$  este de tip întreg. Indicați ce se afișează în urma executării secvenței C/C++ de mai jos.
- ```
void f(int &n)  
{  
    if(n>0) {cout<<n; | printf("%d",n);  
            n=n-1; f(n);  
            cout<<n; | printf("%d",n);  
        }  
}
```
- `x=3; f(x);`
- a. 321000 b. 321012 c. 321123 d. 3210123
4. La balul absolvenților se acordă șase premii pe baza unei tombole cu 30 de bilete, numerotate de la 30 la 1, în această ordine. Utilizând metoda backtracking, se generează toate variantele de a alege biletele celor șase câștigători, care sunt anunțați în ordine descrescătoare a numerelor de pe bilete. Două soluții diferă prin cel puțin un bilet. Primele trei soluții generate sunt, în această ordine: (30, 29, 28, 27, 26, 25), (30, 29, 28, 27, 26, 24), (30, 29, 28, 27, 26, 23). Indicați penultima soluție generată.
- a. (7, 6, 5, 4, 3, 2) b. (7, 6, 5, 3, 2, 1) c. (7, 6, 4, 3, 2, 1) d. (7, 5, 4, 3, 2, 1)
5. Într-un arbore cu 7 noduri, numerotate de la 1 la 7, nodul 3 este rădăcină și pentru orice nod numerotat cu i ($i \in [1, 7]$), dacă se notează cu $F(i)$ numărul de descendenți direcți („fii”) ai lui, atunci $F(i)=0$, dacă i este „frunză”, sau $F(i)=i-1$, în caz contrar. Indicați numărul maxim de „frați” ai nodului 1.
- a. 1 b. 2 c. 3 d. 4

SUBIECTUL al II-lea

(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 ce se afișează în urma executării algoritmului dacă se citește numărul 250326. **(6p.)**
- b. Scrieți două numere din intervalul [1000, 9999] 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 **repetă**...**până când** cu o structură repetitivă cu test inițial. **(6p.)**
2. Un graf neorientat, fără cicluri, are 26 de muchii și două componente conexe. Pentru fiecare dintre cele două componente, scrieți un exemplu de număr de noduri, respectiv de număr de muchii. **(6p.)**
3. Pentru o sală de examen se memorează următoarele date specifice: numărul de locuri disponibile și durata examenului, exprimată în ore. Variabila s memorează datele specifice pentru fiecare dintre cele 10 săli ale unui centru de testare, iar expresiile de mai jos au valori întregi și reprezintă numărul de locuri ale primei săli, respectiv durata examenului din această sală. Definiți o structură C/C++ cu eticheta $sala$, care să permită memorarea datelor pentru o sală de examen, și declarați corespunzător variabila s . **(6p.)**
- $s[0].nrLocuri \quad s[0].durata$

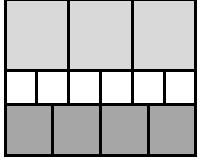
```

citește n
    (număr natural)
x ← 0; y ← 0; p ← 1
repetă
    c ← n % 10; n ← [n/10]
    dacă c % 2 = 0 atunci
        x ← x * 10 + c
        y ← c * p + y; p ← p * 10
    până când n = 0
dacă x < y atunci scrie x
altfel scrie y

```

SUBIECTUL al III-lea

(30 de puncte)

1. Într-un depozit se montează un raft cu trei polițe de aceeași lungime și de lățimi x , y respectiv z . Pe fiecare poliță se plasează, una lângă alta, cutii de forma unui cub cu latura egală cu lățimea poliței. În depozit există suficiente cutii de fiecare tip, iar raftul este echilibrat doar dacă fiecare poliță este ocupată complet de cutii. Subprogramul **depozit** are trei parametri, x , y și z , prin care primește trei numere naturale din intervalul [10, 100], reprezentând lățimile celor trei polițe, exprimate în centimetri. Subprogramul returnează un număr natural, reprezentând lungimea minimă, exprimată în centimetri, a unei polițe a raftului echilibrat. Scrieți definiția completă a subprogramului C/C++.
- Exemplu:** pentru $x=40$, $y=20$ și $z=30$, subprogramul returnează valoarea 120 (raftul are trei polițe, fiecare cu lungime de 120 cm: prima poate conține 3 cutii cu latura de 40 cm, a doua 6 cutii cu latura de 20 cm, iar a treia 4 cutii cu latura de 30 cm). **(10p.)**
- 
2. O metodă de criptare a unui text se bazează pe un cifru format din mai multe coduri, numere naturale. Prin criptare se elimină spațiile, apoi la fiecare pas se ia în considerare o pereche formată dintr-o literă a textului și un cod al cifrului, iar această literă se înlocuiește cu cea obținută prin deplasarea circulară, spre dreapta, în alfabetul limbii engleze, cu un număr de poziții egal cu codul din pereche. Prima pereche este formată din prima literă a textului și primul cod al cifrului, iar după fiecare pereche urmează cea formată din litera următoare a textului, parcurs de la stânga la dreapta, respectiv din codul următor din cifru, parcurs circular, spre dreapta, ca în exemplu. Într-un text de cel mult 100 de caractere cuvintele sunt formate din litere mari ale alfabetului limbii engleze și sunt separate prin unul sau mai multe spații. Alfabetul limbii engleze are 26 de litere. Scrieți un program C/C++ care citește de la tastatură trei numere naturale din intervalul [0, 25], reprezentând, în această ordine, codurile unui cifru, apoi un text de tipul precizat. Programul modifică textul în memorie, criptându-l cu cifrul dat, apoi afișează pe ecran textul obținut.
- Exemplu:** pentru cifrul cu codurile 3, 1, 6 și textul **PIXEL BINE MAPAT** se obține textul **SJDHMHLOKPBVDU** (primele patru perechi sunt P-3, I-1, X-6, E-3). **(10p.)**
3. Fișierul **bac.txt** conține numere naturale din intervalul [1, 10⁹]: pe prima linie două numere, x și y ($x < y$), iar pe a doua linie un șir de cel mult 10⁶ numere, ordonate crescător, separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul de valori pare distincte, din șirul aflat pe a doua linie a fișierului, care aparțin intervalului [x , y]. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
- Exemplu:** dacă fișierul are conținutul
- | | |
|---|--|
| 6 | 11 |
| 1 | 1 1 1 2 2 3 5 5 5 5 6 6 7 8 10 10 12 15 15 21 21 |
- alăturat, pe ecran se afișează 3
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**