

Examenul național de bacalaureat 2025
Proba E. d)
INFORMATICĂ
Limbaajul C/C++

Variantă 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 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.

- Indicați expresia C/C++ cu aceeași valoare ca a expresiei alăturate. 2025%2019+6
a. 2025/2020+5 b. 2025/2021+8 c. 2025%2020+5 d. 2025%2021+8
- Subprogramul **f** este definit alăturat. Indicați ce se afișează în urma apelului de mai jos. **f(3);**

```
void f(int n)
{
    int i;
    for(i=1; i<=n; i++) if(i%2==0) { cout<<i; | printf("%d", i);
                                f(i-1);
                                }
                                else { f(i-1);
                                cout<<i; | printf("%d", i);
                                }
}
```


a. 1211213 b. 123121 c. 123 d. 01201012013
- Indicați o declarație a unui tablou bidimensional **m**, care poate memora maximum 100 de numere reale.
a. **float m[2,50];** b. **float m[4][25];**
c. **float m[10] x float m[10];** d. **int m[100];**
- Utilizând metoda backtracking, s-au generat toate codurile posibile pentru deblocarea unor telefoane, coduri de câte 6 cifre distincte, din mulțimea cifrelor, ordonată crescător. Fiecare cod are primele trei cifre impare și ultimele trei cifre pare. Primele patru coduri sunt 135024, 135026, 135028, 135042. Indicați penultimul cod generat.
a. 957862 b. 957846 c. 975862 d. 975846
- Un graf orientat fără circuite are 10 vârfuri. Indicați numărul maxim de arce ale grafului.
a. 10 b. 45 c. 50 d. 90

SUBIECTUL al II-lea (40 de puncte)

- Algoritmul alăturat este reprezentat în pseudocod.
a. Scrieți ce se afișează în urma executării algoritmului dacă se citește, în această ordine, numerele 7 și 17. (6p.)
b. Dacă pentru variabila **n** se citește valoarea 25, scrieți două numere distincte care pot fi citite pentru variabila **m**, astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze valoarea 25. (6p.)
c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind structura **cât timp...execută** cu o structură repetitivă cu test final. (6p.)
- ```
citește m, n
(numele naturale nenule, m ≤ n)
nr ← 0; i ← m
repetă
 x ← 1
 cât timp x * x < i execută
 x ← x + 1
 dacă x * x = i atunci nr ← i
 altfel i ← i + 1
până când i > n sau nr ≠ 0
scrie nr
```

2. Un graf neorientat cu 6 noduri, numerotate de la 1 la 6, este reprezentat prin listele de adiacență alăturate. Scrieți mulțimea nodurilor și mulțimea muchiilor unui subgraf al acestuia, fără noduri izolate, care să fie graf eulerian. (6p.)
- |    |            |
|----|------------|
| 1: | 2, 3, 4, 6 |
| 2: | 1, 3, 5    |
| 3: | 1, 2, 5    |
| 4: | 1, 5, 6    |
| 5: | 2, 3, 4, 6 |
| 6: | 1, 4, 5    |
3. Variabila **p** memorează simultan, pentru un tip de prăjitură, codul (un număr natural de două cifre), prețul (număr real) și un set de trei numere naturale din intervalul  $[1, 10^2]$ , reprezentând informații specifice, în această ordine: tipul glazurii, tipul cremei principale și numărul de blaturi. Știind că expresiile C/C++ de mai jos au ca valori codul, prețul, respectiv tipul glazurii pentru o prăjitură, scrieți definiția unei structuri cu eticheta **prajitura**, care permite memorarea datelor despre o prăjitură, și declarați corespunzător variabila **p**. (6p.)
- p.cod    p.pret    p.informatii[0]**

### SUBIECTUL al III-lea

(30 de puncte)

1. Numărul natural **an** este **ascendent** al numărului natural **n**, dacă oricare dintre cifrele lui **an** este mai mare sau egală cu cifra unităților lui **n**.  
**Exemplu:** oricare dintre numerele 7, 9, 98 sau 7998 este ascendent al lui 827, dar numărul 857 nu este ascendent al lui 827.  
Subprogramul **ascendent** are trei parametri:  
• **n**, prin care primește un număr natural ( $n \in [0, 10^3]$ );  
• **x** și **y**, prin care primește câte un număr natural din intervalul  $[0, 10^3]$  ( $x < y$ ).  
Subprogramul returnează suma ascendenților lui **n** din intervalul  $[x, y]$ , sau valoarea 0, dacă nu există niciun astfel de ascendent. Scrieți definiția completă a subprogramului C/C++.  
**Exemplu:** dacă **n=827**, **x=9**, **y=800**, subprogramul returnează 7893 ( $9+77+78+79+87+88+89+97+98+99+777+778+779+787+788+789+797+798+799=7893$ ). (10p.)
2. Un **cuvânt semioglindit** se obține dintr-un cuvânt cu  $2 \cdot k$  ( $k \in [1, 10^2]$ ) litere, prin interschimbarea în acesta a secvenței formate din primele **k** litere cu secvența formată din ultimele **k** litere.  
**Exemplu:** din cuvântul platim se obține cuvântul semioglindit timpla.  
Într-un text de cel mult 200 de caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat, pe care îl transformă în memorie, prin înlocuirea fiecărui cuvânt cu număr par de litere, cu cel semioglindit obținut din acesta, ca în exemplu. Programul afișează pe ecran textul obținut, sau mesajul **nu exista**, dacă toate cuvintele au număr impar de litere.  
**Exemplu:** pentru textul am facut fotografii unei flori mari se afișează pe ecran textul ma facut rafiifotog eiun flori rima (10p.)
3. Un tânăr pasionat de călătorii are o listă cu muzee virtuale și, pentru fiecare, câte un singur interval orar, în care acesta poate fi vizitat online, gratuit. Tânărul dispune zilnic de același interval orar pentru vizite; un muzeu este **convenabil** dacă poate fi vizitat online gratuit în timpul disponibil și dacă pentru vizită îi poate aloca cel puțin o oră. Muzeele din listă sunt numerotate cu valori naturale consecutive, începând cu 1, și cel puțin unul este convenabil.  
Fișierul text **bac.in** conține cel mult  $10^5$  linii, iar pe fiecare linie câte o pereche de numere, reprezentând limitele câte unui interval orar: pe prima linie intervalul orar de care tânărul dispune zilnic, iar pe fiecare dintre următoarele linii, intervalul orar de vizitare gratuită pentru câte un muzeu, în ordinea din listă. Limitele intervalelor sunt ore fixe, numere naturale din intervalul  $[8, 22]$ , iar cele aflate pe aceeași linie a fișierului sunt în ordine strict crescătoare și sunt separate printr-un spațiu.  
Se cere să se afișeze pe ecran, separate printr-un spațiu, două valori, reprezentând numărul de muzee convenabile, respectiv numărul de ordine al ultimului astfel de muzeu din lista tânărului. Utilizați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.  
**Exemplu:** dacă fișierul conține valorile alăturate, atunci pe ecran se afișează numerele 3 4.  
(pot fi vizitate trei muzee cu numerele de ordine 1, 2 și 4, în intervalele 16–18, 17–19, respectiv 18–20).  
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)
- |    |    |
|----|----|
| 16 | 19 |
| 15 | 18 |
| 17 | 21 |
| 19 | 21 |
| 18 | 20 |
| 12 | 13 |