

**Examenul național de bacalaureat 2025**

Proba E. d)

**INFORMATICĂ**

Limbajul C/C++

**Varianta 4**

**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.
- Identifierii utilizati 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. Indicați intervalul căruia îi aparține valoarea memorată în variabila **x**, dacă și numai dacă expresia C/C++ alăturată are valoarea 1. **! (x<=2020) && !(x>2025)**  
a. (2020, 2025]      b. (2020, 2025)      c. [2020, 2025]      d. (2021, 2025)
2. Variabila **k** este de tip întreg, iar variabila **s** permite memorarea unui sir de maximum 20 de caractere. Indicați valoarea variabilei **k** în urma executării secvenței alăturate.  
**s = strchr(s, s[1]) - strchr(s, s[3]);**  
a. 9      b. 3      c. 2      d. 1
3. La o probă de dans a unui concurs s-au calificat 6 perechi, notate cu litere distincte din mulțimea ordonată {**A, B, C, D, E, F**}, fiecare pereche prezentând câte un dans. Utilizând metoda backtracking, se generează toate posibilitățile de a stabili ordinea susținerii probei, astfel încât perechea **A** să prezinte prima, iar perechea **F** să prezinte ultima. Primele trei soluții generate sunt (**A, B, C, D, E, F**), (**A, B, C, E, D, F**), (**A, B, D, C, E, F**). Indicați ultima soluție generată.  
a. (A, D, B, C, E, F)      b. (A, E, D, B, C, F)      c. (A, E, D, C, B, F)      d. (A, F, E, D, C, B)
4. Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, are arcele (3, 1), (4, 1), (4, 2), (4, 3), (5, 6), (6, 4). Indicați numărul vârfurilor pentru care gradul exterior este egal cu gradul interior.  
a. 1      b. 2      c. 3      d. 4
5. Într-un arbore cu 50 de noduri, numerotate de la 1 la 50, rădăcina este nodul numerotat cu 1, iar tatăl oricărui alt nod **i** al său este nodul numerotat cu **[i/2]**. Indicați lungimea lanțului elementar cu o extremitate în nodul 32 și cealaltă extremitate în nodul 23.  
a. 5      b. 7      c. 9      d. 11

**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 citesc, în această ordine, numerele **9767** și **8204**. **(6p.)**
- b. Dacă primul număr citit este **2025**, scrieți două valori distincte din intervalul **[10, 99]** care pot fi citite pentru **y** astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, mesajul afișat să fie **DA**, urmat de o valoare numerică. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind prima structură **cât timp...execută** cu o structură repetitivă cu test final. **(6p.)**

citește **x, y**

(numere naturale distincte)

**cx←0; cy←0**

**cât timp **x≥10** sau **y≥10** execută**

**cât timp **x+y≠0** execută**

****cx←cx+x%10; x←[x/10]****

****cy←cy+y%10; y←[y/10]****

**■**

****x←cx; cx←0; y←cy; cy←0****

**■**

****dacă **x=y** atunci scrie "DA ", x****

****altfel scrie "NU ", x, " ", y****

**■**

2. Subprogramul **f** este definit alăturat. Scrieți valorile **f(3, 9)** și **f(1, 1000)**. (6p.)
- ```
int f(int x, int y)
{ if(x*5>y/5) return x;
  return f(x*5, y/5);
}
```
3. Variabila **b** memorează simultan date despre o bijuterie: greutatea acesteia, exprimată în grame, și metalul prețios din care este confectionată (denumirea și prețul unui gram din acel metal). Greutatea bijuteriei și prețul unui gram de metal sunt numere naturale din intervalul  $[1, 10^4]$ , iar denumirea este un sir de maximum 30 de caractere. Știind că expresiile C/C++ de mai jos au ca valori numele metalului din care este confectionată bijuteria, respectiv prețul acesteia, scrieți definiția unei structuri cu eticheta **bijuterie**, care permite memorarea datelor precizate mai sus, și declarați corespunzător variabila **b**.
- b.metal.denumire**      **b.greutate\*b.metal.pret** (6p.)

### SUBIECTUL al III-lea

**(30 de puncte)**

1. Un fermier are un teren de formă dreptunghiulară, pe care vrea să îl împartă în parcele de arie egală, astfel încât numărul de parcele și aria să fie valori naturale, iar numărul de parcele să fie par și strict mai mic decât aria unei parcele. Subprogramul **teren** are doi parametri, **x** și **y**, prin care primește două numere naturale din intervalul  $[2, 100]$ , reprezentând lungimea și lățimea terenului, măsurate în metri. Subprogramul afișează toate soluțiile posibile de împărțire a terenului, fiecare soluție pe câte o linie a ecranului, sub forma a două numere, separate prin mesajul **parcele de arie** și spații adecvate, unde primul număr reprezintă numărul de parcele obținute, iar al doilea aria unei parcele corespunzătoare, ca în exemplu. Dacă nu există nicio astfel de soluție, subprogramul afișează pe ecran mesajul **nu există**. Scrieți definiția completă a subprogramului C/C++.

**Exemplu:** pentru **x=6** și **y=8**, subprogramul afișează pe ecran valorile **2 parcele de arie 24** alăturate, nu neapărat în această ordine (terenul poate fi împărțit și în **3 4 parcele de arie 12** sau **6 parcele de arie 8** parcele de arie **16** sau **8 parcele de arie 6**, dar nu corespund cerinței). (10p.)

2. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 10^2]$ , **m** și **n**, și elementele unui tablou bidimensional cu **m** linii și **n** coloane, numere naturale distincte din intervalul  $[0, 10^9]$ . Programul modifică tabloul în memorie, înlocuind valorile tuturor elementelor aflate pe coloana care conține numărul minim memorat în tablou cu valoarea elementului aflat pe ultima linie și ultima coloană. Tabloul obținut este afișat pe ecran, câte o linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru **m=3, n=4** și tabloul

|    |    |   |    |
|----|----|---|----|
| 17 | 12 | 5 | 19 |
| 10 | 16 | 2 | 4  |
| 11 | 21 | 8 | 9  |

se obține tabloul

|    |    |   |    |
|----|----|---|----|
| 17 | 12 | 9 | 19 |
| 10 | 16 | 9 | 4  |
| 11 | 21 | 9 | 9  |

(10p.)

3. La o centrală solară se monitorizează **stocul zilnic** de energie produsă și **stocul total** calculat pentru fiecare perioadă. Zilele de monitorizare sunt numerotate cu valori naturale consecutive, în ordine cronologică, începând cu ziua 1. O perioadă este formată din cel puțin două zile de monitorizare, consecutive, iar stocul total calculat pentru ea este suma stocurilor zilnice corespunzătoare. O zi este validată dacă stocul zilnic este cel puțin egal cu limita zilnică, **minZ**; o perioadă este validată dacă stocul total calculat pentru ea este cel puțin egal cu limita stabilită pentru perioade, **minP**, fiecare zi a perioadei este validată, iar perioada este maximală în raport cu această proprietate (nu i se mai poate adăuga nicio zi validată).

Fișierul text **bac.in** conține cel mult  $10^6$  numere naturale din intervalul  $[1, 10^3]$ : pe prima linie **minZ** și **minP**, reprezentând limitele precizate pentru validare, iar pe a doua linie stocurile zilnice de energie produse în zile consecutive, în ordinea monitorizării. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran, pentru fiecare perioadă de producție validată, corespunzătoare datelor din fișier, câte un triplet de numere, reprezentând prima și ultima zi a perioadei, respectiv stocul total calculat pentru ea. Valorile din fiecare triplet se afișează pe câte o linie a ecranului, separate prin câte un spațiu, iar dacă nu există nicio astfel de perioadă, se afișează mesajul **nu există**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă fișierul conține valorile alăturate, se afișează pe ecran

|    |    |    |    |    |    |   |   |    |    |    |   |    |    |    |
|----|----|----|----|----|----|---|---|----|----|----|---|----|----|----|
| 10 | 40 |    |    |    |    |   |   |    |    |    |   |    |    |    |
| 65 | 9  | 20 | 25 | 12 | 14 | 7 | 3 | 11 | 15 | 12 | 8 | 19 | 50 | 21 |

3 6 71  
13 15 90

(2p.)

- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)