

Examenul național de bacalaureat 2023  
Proba E. d)  
Informatică  
Limbajul C/C++

Varianta 7

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. Indicați numerele pe care le pot memora variabilele întregi  $x$  și  $y$ , astfel încât valoarea expresiei C/C++ alăturate să fie 1.  $y \% x - (x / y) * 3 != 0$
- a.  $x=25$  și  $y=75$       b.  $x=25$  și  $y=15$       c.  $x=15$  și  $y=0$       d.  $x=10$  și  $y=30$
2. Subprogramul  $f$  este definit alăturat. Indicați valoarea  $f(3, 2)$ .
- ```
int f(int x, int y)
{
    int z;
    if (y==0) return 1;
    z=f(x, y/2);
    if (y%2!=0) return z*z*x;
    return z*z;
}
```
- a. 1      b. 2      c. 9      d. 18
3. Variabila  $s$  poate accesa un șir cu cel mult 20 de caractere. Indicați șirul accesat prin  $s$  în urma executării secvenței alăturate.  $strcpy(s, "2019");$   
 $strcpy(s+3, "23");$
- a. 20123      b. 201923      c. 202223      d. 2023
4. Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte 3 esențe distincte din mulțimea {bergamotă, cireș, iris, lămâie, salcâm}. Două parfumuri sunt distincte dacă diferă prin cel puțin o esență. Primele patru soluții obținute sunt, în această ordine: (bergamotă, cireș, iris), (bergamotă, cireș, lămâie), (bergamotă, cireș, salcâm) și (bergamotă, iris, lămâie). Indicați penultima soluție generată.
- a. (cireș, iris, salcâm)      b. (cireș, lămâie, salcâm)  
c. (lămâie, iris, salcâm)      d. (iris, lămâie, salcâm)
5. Un graf neorientat are 5 noduri, numerotate de la 1 la 5, cu gradele figurate în tabelul alăturat. Indicați o pereche de valori posibile pentru  $x$  și  $y$ .
- |      |   |     |   |   |     |
|------|---|-----|---|---|-----|
| Nod  | 1 | 2   | 3 | 4 | 5   |
| Grad | 2 | $x$ | 3 | 3 | $y$ |
- a. 0 și 4      b. 1 și 5      c. 2 și 3      d. 3 și 7

**SUBIECTUL al II-lea**

**(40 de puncte)**

**1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu  $[c]$  partea întreagă a numărului real  $c$ .

- Scriveți valoarea afișată dacă se citesc, în această ordine, numerele 3, 746, 82, 3067, 67, 78, 178. (6p.)
- Dacă pentru  $n$  se citește numărul 2, scrieți un șir de numere naturale din intervalul  $[0, 9]$  care pot fi citite în continuare, în acea ordine, astfel încât, în urma executării algoritmului, să se afișeze valoarea 0. (6p.)
- Scriveți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scriveți în pseudocod un algoritm, echivalent cu cel dat, înlocuind prima structură **cât timp...execută** cu o structură repetitivă de tip **pentru...execută**. (6p.)

```

citește n (număr natural nenul)
x ← 0
cât timp n > 0 execută
    citește a, b (numere naturale)
    c ← a; p ← 1
    cât timp a > 9 execută
        a ← [a/10]; p ← p * 10
    a ← a * p + b
    dacă a ≠ c atunci
        x ← x + 1
    n ← n - 1
scrie x
    
```

- Într-un arbore cu rădăcină un nod se află pe nivelul  $x$  dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea  $x$ . Pe nivelul 0 se află un singur nod (rădăcina).  
Un arbore cu 6 noduri, numerotate de la 1 la 6, are muchiile  $[1, 2]$ ,  $[2, 3]$ ,  $[2, 6]$ ,  $[3, 4]$ ,  $[3, 5]$ .  
Scrieți nodurile care pot fi desemnate ca rădăcină, astfel încât fiecare dintre arborii obținuți să aibă un număr minim de niveluri. (6p.)

- Variabila  $p$ , declarată alăturat, memorează caracteristicile unui produs: denumire și preț. Scrieți o secvență de instrucțiuni C/C++ prin care variabila  $a$  memorează valoarea primei litere a denumirii produsului respectiv, dacă acesta are prețul strict mai mic decât 100, sau caracterul \* în caz contrar. (6p.)

```

struct produs
{
    char denumire[20];
    int pret;
};
char a;
    
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

- Subprogramul **DNP1** are un singur parametru,  $n$ , prin care primește un număr natural ( $n \in [1, 10^9]$ ), și afișează pe ecran, separați prin câte un spațiu, toți divizorii pozitivi impari ai lui  $n$  care **NU** sunt primi. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă  $n=90$ , se afișează pe ecran, nu neapărat în această ordine, numerele  
1 9 15 45 (10p.)

- Un joc folosește o tablă dreptunghiulară, pe care sunt reprezentate celule de dimensiune egală, dispuse pe  $m$  linii și pe  $n$  coloane. În fiecare celulă este înscris un număr natural.  
Numim **pătrat de valoare p** patru celule ale tablei, situate pe două linii consecutive și pe două coloane consecutive, cu proprietatea că suma valorilor înscrise în aceste celule este egală cu  $p$ . Scopul jocului este determinarea unui pătrat de valoare maximă.

Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ ,  $m$  și  $n$ , apoi elementele unui tablou bidimensional cu  $m$  linii și  $n$  coloane, numere naturale din intervalul  $[0, 10^4]$ , reprezentând numerele înscrise pe tabla de joc, în ordinea dispunerii celulelor corespunzătoare. Programul determină un pătrat de valoare maximă al tablei și afișează pe ecran această valoare.

**Exemplu:** pentru  $m=5$ ,  $n=4$  și tabloul alăturat, se afișează pe ecran valoarea 36, corespunzătoare pătratului evidențiat în figură. (10p.)

|   |    |    |    |
|---|----|----|----|
| 1 | 1  | 2  | 30 |
| 3 | 10 | 1  | 2  |
| 1 | 13 | 12 | 1  |
| 1 | 2  | 3  | 1  |
| 7 | 1  | 20 | 1  |

- Fișierul **date.in** conține pe prima linie două numere naturale din intervalul  $[1, 10^6]$ ,  $m$  și  $n$ , iar pe următoarele două linii numere naturale din intervalul  $[0, 10^2]$ : pe a doua linie un șir **A**, de  $m$  numere, iar pe a treia linie un șir **B**, de  $n$  numere. Numerele aflate pe aceeași linie sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul maxim de perechi de forma  $(p_a, p_b)$  ( $p_a \in [1, m]$ ,  $p_b \in [1, n]$ ), cu proprietatea că termenul de pe poziția  $p_a$  din șirul **A** are aceeași valoare cu termenul de pe poziția  $p_b$  din șirul **B** și că fiecare poziție, corespunzătoare șirului **A**, respectiv șirului **B**, apare în cel mult o pereche, ca în exemplu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele alăturate, se afișează pe ecran 6 (de exemplu, pentru perechile  $(1, 1)$ ,  $(2, 9)$ ,  $(4, 2)$ ,  $(5, 5)$ ,  $(6, 6)$ ,  $(7, 7)$  sau pentru perechile  $(1, 2)$ ,  $(2, 9)$ ,  $(4, 1)$ ,  $(5, 7)$ ,  $(6, 8)$ ,  $(8, 5)$ ).

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