

Examenul național de bacalaureat 2023

Proba E. d)
Informatică
Limbaajul C/C++

Model

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 două valori pe care le poate avea variabila întreagă **x**, astfel încât, pentru fiecare dintre acestea, expresia C/C++ alăturată să aibă valoarea 1. $x \% 20 == x / 23$
a. {20, 40} b. {20, 41} c. {40, 62} d. {60, 83}
- Subprogramul **f** este definit alăturat. Indicați un apel în urma căruia în șirul afișat există două valori egale cu 23.

```
void f(int m, int n)
{ cout<<(m+n)/2<<' '; | printf("%d ", (m+n)/2);
  if(m<=n) f(m+1,n-2);
}
```


a. **f**(20,23); b. **f**(20,26); c. **f**(21,25); d. **f**(21,27);
- Utilizând metoda backtracking, se generează toate torturile formate din trei straturi de tipuri distincte de cremă din mulțimea {castane, mousse, frișcă, șerbet, ganache}. Ultimul strat este de frișcă sau de ganache, iar aceste tipuri de cremă nu pot apărea pe primele două straturi. Două torturi cu straturi din aceleași tipuri de cremă sunt diferite dacă acestea sunt dispuse în altă ordine. Primele patru soluții generate sunt, în această ordine: (castane, mousse, frișcă), (castane, mousse, ganache), (castane, șerbet, frișcă), (castane, șerbet, ganache). Indicați a șasea soluție generată.
a. (mousse, castane, ganache) b. (mousse, șerbet, frișcă)
c. (șerbet, castane, frișcă) d. (șerbet, mousse, ganache)
- Un graf orientat cu 8 vârfuri, numerotate de la 1 la 8, are arcele (1,2), (1,3), (2,3), (2,6), (3,2), (3,4), (3,6), (4,5), (4,7), (6,1), (7,4). Indicați numărul total de vârfuri ale componentei tare conexe din care face parte vârful 3.
a. 2 b. 3 c. 4 d. 5
- Un graf neorientat admite un ciclu eulerian, de lungime 11, precum și un ciclu hamiltonian, de lungime 7. Indicați numărul de muchii care pot fi eliminate din graf, astfel încât graful parțial obținut să fie arbore.
a. 2 b. 3 c. 4 d. 5

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**.

- Scrieți valoarea afișată dacă se citesc, în această ordine, numerele 4, 3, 11 și 25. (6p.)
- Dacă pentru **m**, **n** și **p** se citesc numerele 10, 15, respectiv 2, scrieți două numere care pot fi citite pentru **q** astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afișată să fie 250. (6p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
citește m,n,p,q
  (numere naturale nenule, p≤q)
s←0
pentru x←p,q execută
  | dacă x%m=0 sau x%n=0 atunci
  |   s←s+x
  |
  | dacă x%m=0 și x%n=0 atunci
  |   s←s-x
  |
  |
scrie s
```

- Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **pentru...execută** cu o structură repetitivă cu test inițial. (6p.)

Probă scrisă la informatică

Model

Limbaajul C/C++

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

2. Variabila **c**, declarată alăturat, permite accesul la datele fiecăreia dintre cele trei echipe participante la ultima etapă a unui concurs: numele și rezultatul (număr natural din intervalul $[0,100]$); cele trei echipe au rezultate distincte.

```
struct echipa
{
    char nume[50];
    int rezultat;
};
struct echipa c[3];
```

Scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran, separate prin câte un spațiu, numele celor trei echipe, în ordinea descrescătoare a rezultatelor obținute. (6p.)

3. Variabila **p** este de tip întreg, iar variabilele **s1** și **s2** permit memorarea câte unui șir de cel mult 30 de caractere. Scrieți ce se afișează în urma executării secvenței alăturate. (6p.)

```
strcpy(s1, "plantau fistic");
p=strchr(s1, ' ') - s1;
strcpy(s2, s1+p+1); strcpy(s1+p-1, s2+2);
strcpy(s2+1, s1+2);
cout<<p<<s2; | printf("%d%s", p, s2);
```

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul **DoiTrei** are un parametru, **n**, prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează valoarea 1 dacă toate cifrele lui **n** sunt din mulțimea $\{2, 3\}$ sau valoarea 0 în caz contrar. Scrieți definiția completă a subprogramului.

Exemplu: dacă $n=22323$ sau $n=3$, atunci subprogramul returnează 1, iar dacă $n=2023$ atunci subprogramul returnează 0. (10p.)

2. Suprafața unui loc de joacă pentru copii este împărțită în $n \cdot n$ zone de formă pătrată dispuse pe **n** rânduri și **n** coloane, numerotate de la 1 la **n**, în fiecare zonă fiind plasat un tobogan, un leagăn, un balansoar sau un carusel. Zonelor li se atribuie câte un cod, în funcție de destinație: codul 0 pentru tobogan, codul 1 pentru leagăn, codul 2 pentru balansoar și codul 3 pentru carusel.

Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ($n \in [1, 10^2]$), și cele $n \cdot n$ elemente ale unui tablou bidimensional, reprezentând codurile corespunzătoare zonelor din locul de joacă, în ordinea parcurgerii lor rând după rând, și a fiecărui rând de la stânga la dreapta.

Programul determină zonele destinate plasării câte unui carusel, care NU au nicio latură comună cu o zonă cu tobogan și, pentru fiecare dintre acestea, afișează pe ecran numărul de ordine al rândului pe care se află. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există nicio astfel de zonă, se afișează pe ecran mesajul **nu exista**.

Exemplu: pentru $n=6$ și tabloul alăturat se afișează pe ecran, nu neapărat în această ordine, numerele: 2 3 3 6. (10p.)

2	0	0	2	1	3
3	1	3	1	2	0
2	1	3	3	2	0
0	2	1	3	1	1
3	1	2	0	0	2
0	0	0	2	1	3

3. O pereche de numere naturale (**x**, **y**) se numește **p-ordonată** dacă **x** și **y** au paritate diferită, cu $x > y$, sau dacă **x** și **y** au aceeași paritate.

Exemplu: (7, 4), (8, 7), (7, 9) sunt perechi p-ordonate, dar (7, 8), (8, 9) nu sunt perechi p-ordonate.

Un șir, format din $2 \cdot n$ numere naturale, se numește **p-ordonat** dacă perechile de forma (**x**, **y**) sunt p-ordonate, pentru orice **x**, dintre primii **n** termeni ai șirului, și orice **y**, dintre ultimii **n** termeni ai șirului.

Fișierul **bac.txt** conține numere naturale din intervalul $[0, 10^6]$: pe prima linie un număr nenul, **n**, iar pe a doua linie un șir de $2 \cdot n$ numere, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, în cazul în care șirul aflat în fișier este p-ordonat, sau mesajul **NU**, în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

Exemplu: dacă fișierul are unul dintre conținuturile de mai jos, se afișează pe ecran mesajul **DA**.

5

22 37 21 8 15 2 3 14 4 5

5

22 36 20 8 14 44 42 4 2 5

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

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