

## Examenul național de bacalaureat – decembrie 2022

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

## Simulare județeană

*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 muchie are extremități distincte și oricare două 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.

- Variabilele  $x$  și  $y$  sunt tip întreg. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă valorile întregi nenule memorate în variabilele  $x$  și  $y$  sunt impare.
  - $(x\%2==y\%2) \ \&\& \ (x*y>0)$
  - $(x\%2!=0) \ || \ (y\%2!=0)$
  - $!(x\%2==0) \ || \ !(y\%2==0)$
  - $!(x\%2==0) \ \&\& \ !(y\%2==0)$
- Subprogramul  $f$  este definit alăturat. Indicați valoarea  $f(432115, 0)$ .

```
int f(int n, int p){
    if (n==0) return p;
    else
        return f(n/10, p*10+n%10);
}
```

  - 100016
  - 432115
  - 511234
  - 4321150
- Codul unui seif este alcătuit din toate cifrele care se împart exact la 4, fără ca nicio cifră să se repete. Indicați numărul maxim de încercări necesare până la descoperirea codului.
  - 5
  - 10
  - 23
  - 24
- Un graf neorientat cu 7 noduri, numerotate de la 1 la 7, este reprezentat alăturat prin listele de adiacență. Indicați lungimea maximă a unui ciclu elementar.

1:	2, 4
2:	1, 3, 5, 7
3:	2, 4
4:	1, 3, 6
5:	2, 6
6:	4, 5, 7
7:	2, 6

  - 4
  - 5
  - 8
  - 9
- Un arbore cu 9 noduri, numerotate de la 1 la 9, este reprezentat prin vectorul de tați  $(5, 3, 0, 5, 3, 3, 9, 5, 2)$ . Indicați numărul minim de noduri care trebuie adăugate astfel încât fiecare nod care nu este frunză să aibă un număr par de descendenți direcți.
  - 0
  - 1
  - 3
  - 4

## SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.
- a. Scrieți ce se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 6, 11, 5, 8, 13, 4, 2. (6p.)
- b. Scrieți un set de date de intrare, astfel încât, în urma executării algoritmului să se afișeze valoarea 0. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind structura **pentru...execută** cu o structură repetitivă cu test inițial. (6p.)
2. Pentru o melodie se memorează următoarele informații specifice: numele interpretului, titlul piesei și data lansării. Variabila **v** memorează informațiile specifice pentru fiecare dintre cele 10 melodii lansate într-o lună la o casă de discuri. Scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran titlul melodiilor lansate înainte de anul 2000 pentru care numele interpretului începe cu litera **A**, respectiv mesajul **nu exista** dacă nu sunt melodii care să respecte criteriile date. Declarați eventualele variabile suplimentare utilizate. (6p.)
3. Variabila **i** este de tip întreg, iar variabilele **x**, **a** și **b** permit memorarea a câte unui șir cu cel mult 20 de caractere. Scrieți ce se afișează în urma executării secvenței alăturate. (6p.)
- ```
citește n (număr natural nenul)
nr←0
pentru i←1,n execută
| citește z (număr natural nenul)
| x←1; y←1
| cât timp z>y execută
| aux←x; x←y; y←aux+y
| ■
| dacă z=aux+x atunci nr←nr+1
| ■
| ■
scrie nr

struct data{
    int zi, luna, an;
};
struct melodie{
    char nume[30], titlul[40];
    data data_lansare;
}v[10];

strcpy(a,"albastru");
strcpy(b,"galben"); i=0;
strcpy(x,"");
while (i<strlen(a) && i<strlen(b)){
    if (a[i]<b[i]) strcat(x,"0");
    else strcat(x,"1");
    ++i;
}
cout<<x; | printf("%s",x);
```

## SUBIECTUL al III-lea

(30 de puncte)

1. Un număr natural de 2 cifre se numește "dur" dacă este format cu cifrele de la 1 la 8, iar produsul cifrelor sale este mai mare sau egal cu 18.

Subprogramul `numere_dure` are un singur parametru, `n`, prin care primește un număr natural ( $n \in [1, 100)$ ). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, primele `n` numere dure. Scrieți definiția completă a subprogramului. În cazul în care nu există suficiente numere "dure", se va afișa mesajul "Nu exista".

**Exemplu:** dacă `n=10` atunci, după apel, se afișează pe ecran numerele 36 37 38 45 46 47 48 54 55 56 (10p.)

2. Pe o hartă dreptunghiulară a unui joc pe calculator în care sunt dispuse mai multe încăperi care comunică prin uși montate pe fiecare perete vecin al celei sunt notate locațiile comorilor ce trebuie culese. Jucătorul este teleportat la început într-o cameră. El poate trece dintr-o cameră în camerele care comunică cu aceasta și are la dispoziție doar 4 încercări (N, S, V, E). Jucătorul vrea să știe care este camera care i-ar aduce cel mai mare câștig (colectarea celor mai multe comori, inclusiv din camera în care se situează la un moment dat). Scrieți un program C/C++ care citește, în ordine, de la tastatură, două numere naturale din intervalul  $[2, 10^2]$ , `n` și `m`, apoi elementele unui tablou bidimensional cu `n` linii și `m` coloane, numere din intervalul  $[0, 10]$ , reprezentând numărul de comori din fiecare cameră. Programul afișează pe ecran poziția de pe hartă (linie, coloană), pentru celula care i-ar aduce jucătorului, numărul cel mai mare de comori. Dacă există mai multe soluții se va afișa una dintre ele. Liniile și coloanele se consideră indexate de la 1.

**Exemplu:** pentru `n=4` și `m=6` și tabloul alăturat se vor afișa coordonatele celulei 3, 2 (10p.)

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 3 | 1 | 2 | 4 | 6 |
| 0 | 2 | 8 | 2 | 5 | 8 |
| 4 | 6 | 9 | 1 | 3 | 2 |
| 2 | 5 | 0 | 9 | 2 | 1 |

3. Se numește șir magic, un șir de numere consecutive care pentru fiecare valoare pe care o conține are în componență un număr egal de asemenea valori. Exemplu: șirul 2, 2, 3, 3, 4, 3, 4, 4, 4 este un șir magic deoarece: valoarea 2 apare în șir de 2 ori, valoarea 3 apare în șir de 3 ori iar valoarea 4 apare în șir de 4 ori și numerele care apar în șir fac parte dintr-o secvență de numere consecutive. Fișierul `magice.in` conține cel mult  $10^9$  numere naturale din intervalul  $[1, 10^3)$ . Numerele sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul "Da" dacă numerele din fișier respectă proprietatea unui șir magic și mesajul "Nu" în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de execuție.

Exemplu: dacă fișierul `magice.in` conține numerele:

5 5 2 3 1 4 3 5 3 4 4 5 5 4 2

Se afișează pe ecran mesajul Da

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