



## EXAMENUL DE BACALAUREAT - 2023

### Proba E.d), Simulare județeană Proba scrisă INFORMATICĂ, limbajul C/C++

Filiera teoretică, profil real, specializarea matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- În rezolvările cerute, identificatorii utilizați 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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

#### Subiectul I (20p)

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. Care au fost valorile variabilelor  $x$  și  $y$ , de tip întreg, la începutul executării secvenței de instrucțiuni alăturate dacă la finalul executării  $x$  are valoarea 2007 iar  $y$  are valoarea 2009?

```
x=x-1;  
y=2*x+y;  
x=2*x+1;
```

- a.  **$x=1002$  și  $y=5$**                       b.  **$x=1004$  și  $y=3$**   
c.  **$x=3$  și  $y=1004$**                       d.  **$x=2007$  și  $y=2009$**

2. Se știe că variabila  $a$  de tip întreg memorează valoarea 0. Pentru definiția alăturată a subprogramului `ex`, ce valoare va avea variabila  $a$  în urma executării apelului `ex(10542, 1821, a)`?

```
void ex(int n, int m, int &z)  
{ int c;  
  if (n+m)  
  { c = n%10;  
    if ( m%10>c) c = m%10;  
    z = z*10+c;  
    ex(n/10, m/10, z);  
  }  
}
```

- a. **12500**    b. **11248**  
c. **24811**    d. **2481**

3. Generând toate partițiile mulțimii  $A=\{1,2,3\}$  se obțin următoarele soluții:  $\{1\}\{2\}\{3\}$ ;  $\{1\}\{2,3\}$ ;  $\{1,3\}\{2\}$ ;  $\{1,2\}\{3\}$ ;  $\{1,2,3\}$ . Se observă că prima soluție e alcătuită din exact trei submulțimi. Dacă se folosește exact aceeași metodă pentru a genera partițiile mulțimii  $\{1,2,3,4\}$  stabiliți câte dintre soluțiile generate vor fi alcătuite din exact trei submulțimi.

- a) 3    b) 6    c) 12    d) 5

4. Se consideră un graf neorientat  $G$  cu  $n>2$  noduri și  $n(n-1)/2-1$  muchii. Numărul de vârfuri terminale este:

- a) 0    b) 1    c) cel mult 1    d) cel mult 2

5. Care este numărul grafurilor orientate cu  $n$  noduri cu proprietatea că pentru orice pereche de noduri distincte  $i$  și  $j$  există cel puțin un arc între  $i$  și  $j$ .

- a)  $3^n$     b)  $n!$     c)  $2^n$     d)  $3^{n(n-1)/2}$



**Subiectul II (40p)**

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu  $a \% b$  restul împărțirii numărului  $a$  la numărul  $b$  și cu  $[a]$  partea întreagă a numărului real  $a$ .

a) Scrieți ce se va afișa dacă pentru  $n$  se citește valoarea 31. 6p

b) Determinați o valoare a lui  $n$  pentru care se obține o egalitate între valoarea datei de ieșire și a celei de intrare. 4p

c) Realizați programul C/C++ corespunzător algoritmului dat. 10p

d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască fiecare structură repetitivă cu câte o structură repetitivă condiționată anterior. 8p

```

citește n (număr întreg)
nr ← 0
x ← n
┌ repetă
│ nr ← nr*10+ n % 10
│ n ← [n/10]
└─ pînă când n=0
ok ← 1
┌ pentru j ← 2, [√nr] execută
│   ┌ dacă nr % j=0 atunci
│   │   ok ← 0
│   └─
└─
┌ dacă ok atunci
│   scrie [nr/x]
│   altfel
│   scrie [x/nr]
└─

```

2. Variabila  $c$  memorează simultan date despre 100 de produse: denumirea (șir de caractere maxim 30), prețul (număr real) și data, numere naturale reprezentând ziua, luna și anul expirării produsului. Expresiile C/C++ de mai jos au ca valori denumirea, prețul și ziua, luna și anul expirării celui de al treilea produs. Scrieți definiția unei structuri cu eticheta *produs*, care permite memorarea datelor precizate pentru un produs și declarați corespunzător variabila  $c$ .

```

c[2].denumire c[2].pret c[2].datae.zi c[2].datae.luna
c[2].datae.an

```

(6p.)

3. Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, având inițial toate elementele nenule. Fără a utiliza alte variabile decât cele menționate, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila  $a$  să memoreze tabloul de mai jos.

```

for (i=1; i<=5; i++)
    for (j=1; j<=5; j++)
        .....

```

1	0	0	0	0
0	1	1	1	1
1	0	1	0	0
0	1	0	1	1
1	0	1	0	1

(6p.)



### **SUBIECTUL III (30p)**

1. Subprogramul **divizorik** are doi parametri:

- **n**, prin care primește un număr natural ( $n \in [1, 10^3]$ ) și
- **k**, prin care primește un număr natural ( $k \in [1, 20]$ ).

Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, primele **k** numere naturale mai mari decât **n** care au același număr de divizori ca și **n**. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă **n=20** și **k=3** atunci, după apel, se afișează pe ecran numerele **28 32 44**.

(10p.)

2. Un text are cel mult **100** de caractere și este format din cuvinte, numere naturale, spații și caracterul ,(virgulă). Cuvintele și numerele sunt separate prin cel puțin un spațiu, ca în exemplu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și afișează pe ecran un subsșir din text care conține toate cifrele impare(**1, 3, 5, 7, 9**) și are lungimea minimă, sau mesajul **NU**, în caz contrar. Dacă există mai multe subsșiruri de lungime minimă se va afișa ultimul.

**Exemplu:** pentru textul am 4 ciresi, 19 meri, 315 oi, 72 gaini, 20 cocosi si 13 pui se afișează 9 meri, 315 oi, 7

iar pentru textul in gradina am 12 trandafiri și 15 lalele se afișează **NU**

(10p.)

3. Fișierul **date.txt** conține un șir de cel mult 1000000 numere naturale cu valori de cel mult 6 cifre, separate prin spații. Numim secvență *3putere*, unul sau mai multe numere consecutive ce sunt puteri ale lui 3. De exemplu, 17 21 **3 1 27 3 9 81** 78 56 conține o secvență corectă marcată cu aldine.

Scrieți un program C/C++ care citește numerele din fișier și afișează pe ecran lungimea maximă a unei secvențe *3putere*. Pe a doua linie a ecranului, programul va afișa secvența *3putere* de lungime maximă, valorile fiind separate prin spațiu în ordine crescătoare. Dacă există mai multe secvențe de lungime maximă, se va afișa aceea pentru care suma secvenței este maximă. Alegeți o metodă eficientă ca timp de execuție și ca spațiu de memorie.

**Exemplu1:** Dacă fișierul are conținutul de mai jos

**1 27 9 17 21 3 1 27 3 9 81 78 56 1 3 9 1 81 9**

Se va afișa:

**6**

**1 3 3 9 9 81**

**Exemplu2:** Dacă fișierul are conținutul de mai jos

26 10 17 21 78 56 40

Se va afișa:

**0**

**NU EXISTĂ SECVENȚA**

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

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