

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

Filieră teoretică, profil real, specializare Științe ale naturii

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 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.

SUBIECTUL I

(20 puncte)

I. Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. (20 puncte)

1. Se dau variabilele x și y de tip întreg, x memorând valoarea 4, iar y valoarea 2. Care dintre expresiile C/C++ de mai jos are valoarea 0 (False)?

- a. $x-y!=0$
- b. $x+y>x\%y+1$
- c. $x-2*y==0$
- d. $!(x==2*y)$

2. Care trebuie să fie valoarea variabilei y de tip întreg, astfel încât secvența următoare să execute o singură dată instrucțiunea de afișare

```
int x=10;
do
{
    cout<<"simulare";
    x++;}
while(x>y);
```

- a. 4
- b. 12
- c. 7
- d. 2

3. Stabiliți care este valoarea inițială a variabilei naturale i pentru ca secvența de program de mai jos să afișeze valorile **1 2 3 4 5 6 7 8 9 10**

```
k=1;
for(i=...; i<=2023; i++)
{ cout<<k<<" ";
  k++;
}
```

- a. 4
- b. 34
- c. 123
- d. 2014

4. Variabila x este de tip întreg și memorează un număr natural. Expresia de mai jos are valoarea 0 dacă și numai dacă expresia $x\%5$ are valoarea:

$$((x\%5+1)\%5+1)\%5$$

- a. 2
- b. 7
- c. 3
- d. 4

5. Tablourile unidimensionale a și b au elementele $a=(2, 20, 27, 36, 50)$, iar $b=(3, 5, 8, 45, 63)$. În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:

- a. (2,5,8,36,50)
- b. (2,3,5,8,20,27,36,45,50,63)
- c. (2,3,5,20,8,27,36,45,50,63)
- d. (2,,20,27,36,50,3,5,8,45,63)

SUBIECTUL II

(40 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) Ce valoare va fi afișată dacă se citesc în această ordine
5 222 2043 29 2 20035 **(6p)**
- b) Scrieți un set de date de intrare care să determine
afișarea valorii 9478 **(6p)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat
(10p)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat,
înlocuind adecvat structură repetitivă cât timp cu o
structură repetitivă cu test final **(6p)**

```
citeste n (numar natural nenul)
a ← 0 ;
pentru i ← 1, n executa
    nr ← 1; citeste x;
    cat timp x>9 executa
        nr ← nr*10; x ← [x/10]
    a ← a + x*nr
scrie a
```

2. Pentru a verifica dacă într-un tablou unidimensional de numere întregi având elementele (3,4,7,10,12,17,18,20,46) există un număr întreg x se aplică metoda căutării binare. Știind că numerotarea elementelor, în tablou, se realizează începând cu poziția 0, se cere: **(6p)**

- a) să se afișeze valorile care vor fi comparate pentru a căuta valoarea $x=17$
- b) să se afișeze valorile care vor fi comparate pentru a căuta valoarea $x=5$

3. Variabila **gen** memorează inițiala genului unui film (litera C – dacă genul este comedie, litera D – dacă genul este dramă, litera A – dacă genul este acțiune), variabila **cod** memorează codul filmului (număr natural), iar variabila **an_aparitie** memorează anul apariției filmului (număr natural). Declarați variabilele **gen**, **cod** și **an_aparitie** și scrieți o secvență de instrucțiuni C/C++ care afișează pe ecran codul filmului, în cazul în care este film de comedie, iar în caz contrar afișează anul apariției și genul filmului. **(6p)**

Exemplu: Dacă variabila **gen** memorează litera A, variabila **cod** memorează numărul 5478, iar variabila **an_aparitie** memorează numărul 2004 se afișează

2004 Actiune

1. Se citește un număr natural, n ($n \geq 2$), și se cere să se scrie numărul din intervalul închis $[2, n]$ care are cei mai mulți factori primi. Dacă există mai multe numere cu această proprietate, se scrie cel mai mare dintre ele. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate.

Exemplu: dacă $n=100$, se scrie **90**.

(10p.)

2. Scrieți un program C/C++ care citește de la tastatură un număr natural n ($2 < n < 50$) și construiește în memorie un tablou unidimensional, astfel încât parcurgându-l de la stânga la dreapta să se obțină șirul primelor n numere naturale, impare, care **NU** sunt divizibile cu **3**, ordonat strict descrescător. Programul afișează pe ecran elementele tabloului obținut, separate prin câte un spațiu.

Exemplu: pentru $n=7$, se obține tabloul (19, 17, 13, 11, 7, 5, 1).

(10p.)

3. Fișierul bac.txt conține pe prima linie un număr natural cu cel mult nouă cifre, x , iar pe a doua linie un șir de cel puțin două și cel mult **1000000** de numere naturale cu cel mult nouă cifre. Numerele din șir sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran penultimul termen al șirului care are ultima cifră egală cu prima cifră a numărului x . Dacă în șir nu există o astfel de valoare, pe ecran se afișează mesajul **Nu exista**.

Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

Exemplu: dacă fișierul bac.txt conține numerele

12

345 8911 1245 51 67123 931 1234578

atunci pe ecran se afișează **51**.

(10p.)