

Examenul național de bacalaureat 2023

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

Model

Filieră teoretică, profil real, specializare științe ale naturii

- 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.

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 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}
2. Variabilele i și j sunt de tip întreg. Indicați cu ce se pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran, în această ordine, numerele de mai jos.
- ```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```
- ```
for(i=1;i<=5;i++)
{ for(.....)
  cout<<j<<' '; | printf("%d ",j);
  cout<<endl; | printf("\n");
}
```
- a. $j=1; j>=i; j--$ b. $j=5; j>=1; j--$
- c. $j=6-i; j>=1; j--$ d. $j=i; j>=1; j--$
3. Tablourile unidimensionale A și B au valorile $A=(23,20,17,9,6)$ și $B=(50,16,14,5,3)$ și sunt interclasate în ordine crescătoare, fiind parcurse de la **dreapta la stânga**. Indicați numărul elementelor tabloului B care se compară cu elementul cu valoarea 17 din tabloul A pe parcursul aplicării metodei.
- a. 1 b. 2 c. 3 d. 4
4. Indicați intervalul căruia îi aparține valoarea variabilei întregi x , astfel încât expresia C/C++ alăturată să aibă valoarea 1. $abs(x-10)<2023$
- a. (-2013,2033) b. (-2023,2033) c. [-2013,2023] d. (-2033,2033)
5. În secvența alăturată toate variabilele sunt de tip întreg ($n>1$). Indicați cu ce pot fi înlocuite expresiile A și B astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran toți divizorii primi ai lui n .
- ```
d=2;
while(d*d<=n)
{ p=0;
 while(n%d==0){ n=n/d; p=p+1; }
 if(p!=0) cout<<d<<" "; | printf("%d ",d);
 d=d+1;
}
if(A) cout<<B; | printf(printf("%d ",B);
```
- a. A:  $p>0$                                       b. A:  $p>1$                                       c. A:  $n>0$                                       d. A:  $n>1$   
B: d                                              B: n                                              B: d                                              B: n

**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  $b$ .

- a) Scrieți valoarea afișată dacă se citesc, în această ordine, numerele 4, 3, 11 și 25. **(6p.)**
- b) 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.)**
- c) 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

```

- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **pentru...execută** cu o structură repetitivă cu test inițial. **(6p.)**
2. Pentru fiecare echipă participantă la un concurs se memorează date specifice: indicativul (o literă mare a alfabetului englez) și rezultatul (număr natural din intervalul  $[0,100]$ ). Variabilele  $i1$  și  $r1$  memorează indicativul și rezultatul primei echipe, variabilele  $i2$  și  $r2$  memorează indicativul și rezultatul celei de a doua echipe, iar variabilele  $i3$  și  $r3$  memorează indicativul și rezultatul celei de a treia echipe. Cele trei echipe au rezultate distincte. Declarați variabilele precizate și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran, separate prin câte un spațiu, indicativele celor trei echipe, în ordinea descrescătoare a rezultatelor obținute. **(6p.)**
3. Pentru a verifica dacă în tabloul unidimensional (3, 7, 10, 12, 20, 23, 65) există elementul cu valoarea  $x=21$ , se aplică metoda căutării binare. Scrieți succesiunea de elemente din tablou ale căror valori se compară cu valoarea lui  $x$  pe parcursul aplicării metodei indicate. **(6p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Se citește un număr natural,  $n$ , și se cere să se scrie valoarea 1 dacă toate cifrele lui  $n$  sunt din mulțimea  $\{2, 3\}$  sau valoarea 0 în caz contrar.  
**Exemplu:** dacă  $n=22323$  sau  $n=3$ , atunci se scrie 1, iar dacă  $n=2023$  atunci se scrie 0.  
Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **(10p.)**

2. Suprafața unui loc de joacă pentru copii este împărțită în  $n$  zone de formă dreptunghiulară dispuse pe  $n$  rânduri, 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  $n$  elemente ale unui tablou unidimensional, reprezentând codurile corespunzătoare zonelor din locul de joacă, în ordinea parcurgerii lor rând după rând. 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  $n=8$  și tabloul alăturat, cu elemente numerotate de sus în jos, se afișează pe ecran, nu neapărat în această ordine, numerele: 2 6 8. **(10p.)**

|   |
|---|
| 1 |
| 3 |
| 2 |
| 0 |
| 3 |
| 3 |
| 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), (7, 9) sunt perechi p-ordonate, dar (7, 8) nu este pereche p-ordonată.

Un șir, format din  $2 \cdot n$  numere naturale, se numește **ip-ordonat** dacă primii  $n$  termeni sunt pari sau dacă toate perechile de forma  $(x, y)$  sunt p-ordonate, pentru orice  $x$  **impar**, 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 ip-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 43 14 4 5

5  
22 36 20 8 14 44 43 4 2 5

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