Examenul național de bacalaureat 2026
Proba E. d)
INFORMATICĂ
Limbajul C/C++

Simulare

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. Variabila **x** este de tip întreg. Indicați o expresie care are valoarea **1** dacă și numai dacă expresia C/C++ alăturată are valoarea **1**.

```
x<=20 || x>26
```

a. `!(x>20) || !(x<=26)` **b.** `!(x>=20) && !(x>=26)`
c. `!(x<20 || x<=26)` **d.** `!(x<20 && x<26)`

2. Pentru a verifica dacă în tabloul unidimensional `(5,10,14,16,19,21,26)` există elementul cu valoarea **x**, număr natural, se aplică metoda căutării binare. Indicați mulțimea tuturor valorilor posibile ale lui **x**, astfel încât succesiunea de elemente ale tabloului a căror valoare se compară cu valoarea lui **x** pe parcursul aplicării metodei indicate să fie **16**, **21**, **19**.

a. `{19}` **b.** `{18,20}` **c.** `{18,19,20}` **d.** `{17,18,19,20}`

3. Indicați valoarea expresiei C/C++ alăturate.

```
abs(25-3-2026)
```

a. `-2054` **b.** `-2004` **c.** `2004` **d.** `2054`

4. O probă a unui examen începe la ora **9:00**, iar variabila întreagă **p** memorează durata probei, în minute. Indicați secvența C/C++ care atribuie variabilelor întregi **h** și **m**, ora, respectiv minutul la care se încheie proba, în aceeași zi.

a. `h=9+p%60; m=p/60;` **b.** `h=9+p/60; m=p%60;`
c. `h=(9+p)%24; m=p/60;` **d.** `h=(9+p)/24; m=p%60;`

5. La balul absolvenților se acordă trei premii pe baza unei tombole cu **10** bilete, numerotate de la **1** la **10**. Variabilele **i**, **j** și **k** sunt de tip întreg, iar secvența de instrucțiuni C/C++ de mai jos generează toate variantele de a alege biletele celor trei câștigători, care sunt anunțați în ordine descrescătoare a numerelor de pe bilete. Două soluții diferă prin cel puțin un bilet. Indicați penultima soluție generată.

```
for(i=10;i>=3;i--)
  for(j=i-1;j>=2;j--)
    for(k=j-1;k>=1;k--)
      cout<<i<<" "<<j<<" "<<k<<endl; | printf("%d %d %d\n",i,j,k);
```

a. (`5,2,1`) **b.** (`4,3,2`) **c.** (`4,3,1`) **d.** (`4,2,1`)

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** și cu **[c]** partea întreagă a numărului real **c**.

```
citeşte n
   (număr natural)
 x←0; y←0; p←1
┌repetă
| c←n%10; n←[n/10]
|┌dacă c%2=0 atunci
|| x←x*10+c
|| y←c*p+y; p←p*10
|└■
└până când n=0
┌dacă x<y atunci scrie x
|altfel scrie y
└■
```

a. Scrieți ce se afișează în urma executării algoritmului dacă se citește numărul **250326**. **(6p.)**

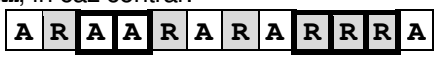
b. Scrieți două numere din intervalul **[1000,9999]** care pot fi citite, astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea **64**. **(6p.)**

c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura **repetă...până când** cu o structură repetitivă cu test inițial. **(6p.)**

2. Tablourile unidimensionale **A=(26,19,16,8,4)** și **B=(50,15,12,3,1)** sunt interclasate în ordine crescătoare, fiind parcurse de la **dreapta la stânga**. Scrieți toate valorile elementelor tabloului **B** care se compară cu elementul cu valoarea **16** din tabloul **A** pe parcursul aplicării metodei. **(6p.)**

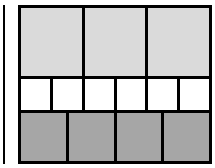
3. Un mozaic este alcătuit din mai multe plăcuțe de formă pătrată, cu dimensiuni egale, de culoare albă sau roșie, așezate pe un rând și aliniate ca în exemplu. Mozaicul are un model alternativ dacă oricare două plăcuțe cu o latură comună au culoare diferită.
Variabilele **x1**, **x2** și **x3** memorează câte o literă corespunzătoare primelor trei plăcuțe din mozaic, în această ordine, notându-se cu litera **R** cele roșii și cu litera **A** cele albe. Declarați variabilele **x1**, **x2** și **x3** și scrieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze mesajul **alternativ**, dacă cele trei plăcuțe respectă modelul alternativ al mozaicului, sau mesajul **neconform**, în caz contrar.
Exemplu: pentru mozaicul alăturat, unde sunt evidențiate două zone în care nu se respectă modelul, variabilele **x1**, **x2** și **x3** memorează literele **A**, **R**, respectiv **A**, și se afișează pe ecran mesajul **alternativ**



(6p.)

SUBIECTUL al III-lea **(30 de puncte)**

1. Într-un depozit se montează un raft cu trei polițe de aceeași lungime și de lățimi **x**, **y** respectiv **z**. Pe fiecare poliță se plasează, una lângă alta, cutii de forma unui cub cu latura egală cu lățimea poliței. În depozit există suficiente cutii de fiecare tip, iar raftul este echilibrat doar dacă fiecare poliță este ocupată complet de cutii.
Se citesc trei numere naturale din intervalul **[10,100]**, **x**, **y** și **z**, reprezentând lățimile celor trei polițe, exprimate în centimetri, și se cere să se scrie un număr natural, reprezentând lungimea minimă, exprimată în centimetri, a unei polițe a raftului echilibrat.
Scrieţi, în pseudocod, algoritmul de rezolvare pentru problema enunțată.
Exemplu: pentru **x=40**, **y=20** și **z=30**, se scrie valoarea **120** (raftul are trei polițe, fiecare cu lungime de **120 cm**: prima poate conține **3** cutii cu latura de **40 cm**, a doua **6** cutii cu latura de **20 cm**, iar a treia **4** cutii cu latura de **30 cm**). **(10p.)**



2. O metodă de criptare a unui șir de numere se bazează pe un cifru format din mai multe coduri, numere naturale. Prin criptare, la fiecare pas se ia în considerare o pereche formată dintr-un termen nenul al șirului și un cod al cifrului, iar acest termen este înlocuit cu suma dintre el și codul din pereche. Prima pereche este formată din primul termen nenul al șirului și primul cod al cifrului, iar după fiecare pereche urmează cea formată din termenul următor nenul al șirului, parcurs de la stânga la dreapta, respectiv din codul următor din cifru, parcurs circular, spre dreapta, ca în exemplu.
Scrieţi un program C/C++ care citeşte de la tastatură numere naturale: un număr **n** (**n∈[1,100]**), apoi un șir de **n** numere din intervalul **[0,10⁴]**, elemente ale unui tablou unidimensional, iar la final trei numere din intervalul **[0,10]**, reprezentând, în această ordine, codurile unui cifru. Programul modifică tabloul în memorie, criptând șirul cu cifrul dat, apoi afișează pe ecran tabloul obținut.
Exemplu: pentru **n=11**, tabloul (**9,6,8,3,1,0,21,14,0,20,5**) și cifrul cu codurile **3**, **1**, **6** se obține tabloul (**12,7,14,6,2,0,27,17,0,21,11**); primele patru perechi sunt **9–3**, **6–1**, **8–6**, **3–3**. **(10p.)**

3. Fișierul **bac.txt** conține numere naturale din intervalul **[1,10⁹]**: pe prima linie un număr, **x**, iar pe a doua linie un șir de cel mult **10⁶** numere, ordonate crescător, separate prin câte un spațiu.
Se cere să se afișeze pe ecran numărul de valori pare distincte din șirul aflat pe a doua linie a fișierului care sunt mai mari sau egale cu **x**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
Exemplu: dacă fișierul are conținutul alăturat, pe ecran se afișează **4**

```
5
1 1 1 2 2 3 5 5 5 5 6 6 7 8 10 10 12 15 15 21 21
```

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