

Examenul național de bacalaureat 2024

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. Indicați valoarea expresiei C/C++ alăturate. 6.3/20+24

- a. 0 b. 24 c. 24.315 d. 24.9

2. În secvența de mai jos toate variabilele sunt de tip întreg.

```
for(i=0; i<5; i++)  
{ for(j=0; j<5; j++)  
    if(j<i) cout<<.....<<" "; | printf("%d ", .....);  
    else cout<<2*i<<" "; | printf("%d ", 2*i);  
    cout<<endl; | printf("\n");  
}
```

0 0 0 0 0
0 2 2 2 2
0 2 4 4 4
0 2 4 6 6
0 2 4 6 8

Indicați o expresie care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran valorile din figura de mai sus, în această ordine.

- a. 2+j b. 2*j c. 2+i+j d. 2*i*j

3. Elementele unui tablou unidimensional sunt, în această ordine, (3,6,10,16,20,24,49). Pentru a verifica dacă în tablou există elementul cu valoarea **x=22**, se aplică metoda căutării binare. Indicați succesiunea de elemente cu care se compară valoarea **x** pe parcursul aplicării metodei indicate.

- a. 16,6,24 b. 16,24,20 c. 49,24,20 d. 3,10,20,49

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

floor(sqrt(24))

- a. 25 b. 24 c. 5 d. 4

5. În secvența de mai jos toate variabilele sunt de tip întreg și memorează numere naturale.

Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **y** să memoreze valoarea **2ⁿ** (**n** ∈ [0,15]).

```
y=1; x=2;  
while(n>0)  
    if(.....){ x=x*x; n=n/2; }  
    else { y=y*x; n=n-1; }
```

- a. n%2==0 b. n/2>=1 c. n%4>2 d. n/4<2

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**.

a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește numărul **32625**. (6p.)

b. Scrieți două numere din intervalul **[10,100]** care pot fi citite pentru variabila **n**, astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea **2**. (6p.)

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

d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat a **doua** structură **cât timp...execută** cu o structură repetitivă cu test final. (6p.)

```
citește n  
    (număr natural nenul)  
s←0; d←2  
cât timp d*d≤n execută  
    p←0  
    cât timp n%d=0 execută  
        n←[n/d]; p←1  
    s←s+d*p; d←d+1  
dacă n≠1 atunci  
    s←s+n  
scrie s
```

2. Un șir se obține prin interclasarea crescătoare a elementelor din tablourile unidimensionale $A=(2, 3, 7, 11, 20)$ și $B=(3, 5, 6, 10, 16, 24)$, parcurse de la stânga la dreapta, la fiecare pas avându-se în vedere pentru comparare câte o pereche (x, y) , unde x este un element din A , iar y este un element din B . Scrieți perechile avute în vedere pentru comparare la al cincilea, respectiv la al șaselea pas, în ordinea prelucrării. (6p.)
3. Pentru un joc video se memorează categoria din care face parte (litera A , dacă este un joc de acțiune sau litera S dacă este un joc de strategie) și numărul de niveluri de dificultate (număr natural din intervalul $[1, 10]$). Variabilele $c1$ și $n1$ memorează categoria, respectiv numărul de niveluri de dificultate ale unui joc video, iar variabilele $c2$ și $n2$ memorează categoria, respectiv numărul de niveluri de dificultate ale altui joc video. Declarați corespunzător variabilele $c1$ și $c2$ și, știind că cele două jocuri au numere diferite de niveluri de dificultate, scrieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze pe ecran datele jocului cu mai multe niveluri: denumirea completă a categoriei (**ACȚIUNE** sau **STRATEGIE**), urmată de un spațiu și apoi de numărul de niveluri de dificultate ale jocului.
Exemplu: dacă $c1$ memorează litera A și $n1=2$, iar $c2$ memorează litera S și $n2=4$, atunci se afișează pe ecran **STRATEGIE 4** (6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Se citește un număr natural, n ($n \in [1, 10^4]$), și se cere să se scrie numărul obținut prin duplicarea fiecărei cifre impare a lui n sau -1 , dacă acesta nu are nicio cifră impară. Scrieți în pseudocod algoritmul corespunzător.
Exemplu: dacă $n=3361$, se scrie numărul **3333611**. (10p.)
2. Un **șablon** este un șir de numere întregi, în care termenii strict negativi sunt numiți **termeni generici**. Un computer generează un șir pe baza unui astfel de șablon, prin înlocuirea fiecărui termen generic al șablonului cu unul dintre numerele cu aceeași cifră a unităților, preluat dintr-o **listă** dată de numere naturale.
Exemplu: termenul generic -20 se poate înlocui cu 0 sau cu 50 , dacă lista este $(24, 2, 0, 36, 201, 50)$. Scrieți un program C/C++ care citește de la tastatură un număr natural, nL ($nL \in [1, 100]$), și cele nL elemente ale unui tablou unidimensional L , apoi un număr natural, nS ($nS \in [1, 100]$), și cele nS elemente ale unui tablou unidimensional S . Tabloul L reprezintă o listă de tipul precizat mai sus și elementele sale sunt numere naturale din intervalul $[0, 10^3]$, iar tabloul S reprezintă un șablon de tipul precizat mai sus și elementele sale sunt numere întregi din intervalul $[-10^9, 10^9]$. Programul transformă în memorie tabloul corespunzător șablonului, astfel încât acesta să memoreze unul dintre șirurile care pot fi generate de computer pe baza acestui șablon și a listei date și apoi afișează pe ecran, separate prin câte un spațiu, elementele tabloului obținut sau mesajul **imposibil** dacă nu se poate genera un astfel de șir.
Exemplu: dacă $nL=7$, lista este $L=(24, 2, 0, 36, 21, 104, 50)$, iar $nS=8$
pentru șablonul $S=(-16, 0, 3, -452, -10, -974, 10, -20)$
se generează șirul $(36, 0, 3, 2, 0, 24, 10, 0)$
sau șirul $(36, 0, 3, 2, 0, 104, 10, 50)$
etc., iar pentru șablonul $S=(-17, 0, 3, -452, -10, -975, 10, -20)$
se afișează mesajul **imposibil** (10p.)
3. Un șir se numește de tip **api** dacă numărul de apariții ale fiecărui termen are o paritate egală cu a acestuia. Fișierul **bac.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[1, 10^3]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, dacă șirul este de tip **api**, sau mesajul **NU** în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
Exemplu: dacă fișierul conține numerele **2 27 8 2 27 2 2 14 14 8 27** se afișează pe ecran **DA** (termenul par **2** apare de **4** ori, **4** fiind tot număr par, termenii pari **8** și **14** apar de câte **2** ori, **2** fiind tot număr par, iar termenul impar **27** apare de **3** ori, **3** fiind tot număr impar).
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)