

Simulare pentru Examenul de bacalaureat ianuarie 2026

Proba E. d)
Informatică

Simulare 2

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)

1. Se consideră variabilele:

`int a = 7, b = 3, c = 5;`

`float x = 2.5, y = -1.0;`

Indicați rezultatul expresiei: `(x + a % c) / (float)(b + y) + a / b * c`

- a. 11 b. 11.75 c. 12 d. 12.25

2. Variabilele `i` și `j` sunt de tip întreg. Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, rezultatul afișat să fie cel din exemplul alăturat.

`for (i=0; i<6; i++)`

`{`

`for (j=0; j<6; j++)`

`cout << ... << " ";`

`cout << endl;`

`}`

0 1 2 0 1 2

1 2 0 1 2 0

2 0 1 2 0 1

0 1 2 0 1 2

1 2 0 1 2 0

2 0 1 2 0 1

- a. `(i+j)` b. `(i+j) % 2` c. `(i+j) % 2 + 1` d. `(i+j) % 3`

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

a. 4

b. 5

c. 4.47214

d. 20

`ceil(sqrt(20))`

4. În secvența alăturată toate variabilele sunt de tip întreg (`n>1`). Indicați expresia cu care pot fi înlocuite punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila `ok` să aibă valoarea 1 dacă și numai dacă numărul memorat în variabila `n` este prim.

a. `d+d<n`

b. `d+d<=n`

c. `d*d<=n`

d. `d*d<n`

`ok=1; d=2;`

`while(ok==1 &&)`

`if(n%d==0) ok=0;`

`else d=d+1;`

5. Interclasând descrescător tablourile unidimensionale `(5,9,13,16,20)` și `(19,16,9,7,3,1)` se obține:

a. `(20,19,16,16,13,9,9,7,5,3,1)`

b. `(20,19,16,13,9,7,5,3,1)`

c. `(19,16,9,7,3,1,20,16,13,9,5)`

d. `(19,16,9,7,3,1,20,13,5)`

SUBIECTUL al II - lea

(40 de puncte)

1. Se consideră algoritmul alăturat, 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 dacă se citesc, în această ordine, numerele 7, 55, 72, 31, 17, 100, 93, 2. (6p.)

b) Dacă pentru `n` se citește valoarea 3, scrieți un set de numere distincte din intervalul `(0, 102)` care pot fi citite în continuare, astfel încât, în urma executării algoritmului, să se afișeze valoarea 0. (6p.)

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

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

citește `n` (număr natural nenul)

`m` ← 0

pentru `i` ← 1, `n` **execută**

citește `x` (număr natural nenul)

`d` ← 2

cât timp `d` ≤ `[x/d]` și `x%d` ≠ 0 **execută**

`d` ← `d+1`

`■`

dacă `d` ≤ `[x/d]` și `x > m` **atunci**

`m` ← `x`

`■`

scrie `m`

2. În secvența alăturată toate variabilele sunt de tip întreg, iar de la tastatură se citesc 10 numere naturale.
Scrieți secvența înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, valoarea variabilei **maxImp** să fie egală cu cel mai mare dintre numerele citite care sunt impare și au exact trei cifre, sau să fie egală cu -1 dacă nu există astfel de numere.

```
maxImp=.....;  
for (i=1;i<=10;i++)  
{ cin>>x;  
.....  
}
```

(6p.)

3. Se consideră un tablou unidimensional în care elementele sunt, în această ordine, (4, 7, 10, 12, 15, 21, 49). Pentru a verifica dacă în tablou există elementul **x** cu valoarea 16, se aplică metoda căutării binare.

Precizați succesiunea corectă de elemente din tablou a căror valoare se compară cu valoarea lui **x** pe parcursul aplicării metodei indicate.

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Un număr natural se numește **aranjat** dacă are cel puțin o cifră pară, cel puțin o cifră impară și toate cifrele sale pare respectiv impare sunt grupate (apar pe poziții alăturate în scrierea numărului). De exemplu 71324, 56, 4813 sunt numere aranjate, dar 135 sau 826 sau 1234 nu sunt numere aranjate.

Se citește un număr natural, **n** ($n \geq 10$), și se cere să se scrie valoarea 1 dacă numărul **n** este aranjat sau valoarea 0 în caz contrar. Scrieți, în pseudocod, algoritmul de rezolvare pentru problema enunțată.

Exemplu: dacă $n=7152$ sau $n=66233$, algoritmul va afișa valoarea 1, iar dacă $n=51234$ sau $n=1759$ algoritmul va afișa valoarea 0.

(10p)

2. Un **număr marcat** este un număr de cel puțin trei cifre care începe și se termină cu aceeași cifră. De exemplu 1561 este un număr marcat. Un tablou unidimensional cu cel mult 200 de elemente conține numere naturale de cel mult nouă cifre, unele dintre ele fiind numere marcate.

Scrieți un program C/C++ care citește de la tastatură **n**, un număr natural ($n \leq 200$) și cele **n** elemente ale unui tablou unidimensional de tipul precizat și transformă tabloul în memorie astfel încât:

- fiecare număr marcat este înlocuit cu numărul obținut prin eliminarea primei și a ultimei cifre
- fiecare număr care nu este marcat și este număr par este înlocuit cu 0
- fiecare număr care nu este marcat și este număr impar este înlocuit cu 1

Programul afișează pe ecran tabloul obținut.

(10p)

Exemplu: pentru $n=7$ și tabloul: 123 565 12 44 6896 333 597

se afișează pe ecran: 1 6 0 0 89 3 1

3. La un centru de testare sunt conectate, în ordine, mai multe device-uri inteligente. Fiecare device transmite o singură cifră (0–9), reprezentând nivelul bateriei în momentul testării: 0 dacă bateria este descărcată, sau o cifră nenulă reprezentând nivelul bateriei (cu cât cifra este mai mare, cu atât bateria este mai încărcată). Fișierul **bac.txt** conține cel mult 10^6 astfel de cifre, separate prin câte un spațiu. Se cere să se afișeze pe ecran toate nivelurile distincte de baterie care apar imediat după un device descărcat (0). Nivelurile determinate se vor afișa în ordine strict crescătoare, separate prin câte un spațiu. Dacă nu există astfel de niveluri, se va afișa mesajul: **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul conține valorile: 3 0 7 1 0 4 4 0 9 2 0 2 0 0 8 0 7 atunci pe ecran se vor afișa valorile 2 4 7 8 9, deoarece valorile care apar imediat după fiecare apariție a lui 0 sunt: 7, 4, 9, 2, 0, 8, 7.

Dintre acestea, nivelurile valide (1–9) distincte sunt: 2 4 7 8 9.

Dacă fișierul conține valorile: 7 5 9 6 3 4 7 8 9 1 2 sau valorile 0 0 0 atunci pe ecran se va afișa **nu exista**.

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia.

(2p)

b. Scrieți programul C/C++ corespunzător algoritmului proiectat.

(8p)