

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

Simulare

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

- 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.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

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. Variabilele **a** și **b** sunt de tip **int**, iar variabilele **c** și **d** sunt de tip **float**. Care dintre următoarele instrucțiuni de atribuire nu este corectă din punct de vedere sintactic?

- a. **c=d+2*a**; b. **a=(d<=c)**; c. **d=2-c%2**; d. **c=sqrt(b*b)**;

2. Subprogramul **f** este definit alăturat.

Indicați ce se afișează în urma apelului de mai jos.

f(5);

void f (int x)

```
{  
    while(x>1)  
    {  
        x=x-1;  
        f(x-1);  
    }  
    cout<<x;  
}
```

- a. **11111111** b. **01010101** c. **10101101** d. **00000000**

3. Utilizând metoda backtracking se generează toate posibilitățile de a scrie numărul **10** ca sumă de numere prime. Soluțiile generate sunt, în această ordine: **2+2+2+2+2**, **2+2+3+3**, **2+3+5**, **3+7**, **5+5**. Folosind aceeași metodă se generează toate posibilitățile de a scrie numărul **9** ca sumă de numere prime. Indicați a treia soluție generată.

- a. **2+7** b. **2+2+5** c. **2+2+2+3** d. **3+3+3**

4. Pentru arborele cu rădăcină, cu **9** noduri, numerotate de la **1** la **9**, având următorul vector de „tați” **T=(8,7,6,6,7,7,8,0,8)**, care este numărul de descendenți ai nodului **7**?

- a. **3** b. **2** c. **1** d. **5**

5. Un graf orientat are 9 vârfuri și arcele: [1,3], [2,1], [3,2], [4,5], [5,6], [6,4], [7,9], [8,7], [9,8]. Care este numărul maxim de arce ce pot fi adăugate astfel încât în graful obținut să avem același număr de componente tare conexe ca în graful inițial.

- a. 9 b. 36 c. 0 d. 27

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

a. Scrieți ce se afișează în urma executării algoritmului dacă se citesc, în această ordine, valorile 12 și 48. (6p)

b. Scrieți două perechi de valori distincte din intervalul [10,100] care pot fi citite pentru variabilele a și b astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afișată să fie 9. (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 repetitivă cu o structură repetitivă cu test final. (6p)

```

citește a,b (numere naturale
nenule)
p←1
d←2
cât_timp d≤a și d≤b execută
    n←0
    cât_timp a%d=0 și b%d=0 execută
        p←p*d
        n←n+1
        a←[a/d]
        b←[b/d]
    sfarsit_cât_timp
    dacă n%2=1 atunci
        p←[p/d]
    Sfârșit_dacă
    d←d+1
sfarsit_cât_timp
scrie p

```

2. Scrieți listele de adiacență prin care este reprezentat un exemplu de graf neorientat conex, cu 7 noduri, numerotate de la 1 la 7, care este eulerian, dar nu este hamiltonian. (6p)

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 7 linii și 6 coloane, numerotate începând de la 0, având inițial toate elementele nule.

Scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie cu instrucțiuni adecvate, dintre care cel mult patru de atribuire, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat.

```

for(i=0;i<7;i++)
    for(j=0;j<6;j++)
        .....

```

(6p)

1	2	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30
36	35	34	33	32	31
37	38	39	40	41	42

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

1. (10p.) Subprogramul **sub** are doi parametri: **v**, prin care primește un tablou unidimensional cu maximum 111 elemente numere întregi și **n**, numărul de elemente din tablou. Subprogramul returnează poziția celui mai mare număr "călduros" din tablou. Dacă sunt mai multe astfel de numere, subprogramul va returna poziția primului cel mai mare astfel de număr. Dacă nu există astfel de numere subprogramul va returna -1.

Un număr întreg este "călduros" dacă prima și ultima lui cifră au aceeași paritate.

Scrieți în limbajul C/C++ definiția completă a subprogramului **sub**.

Exemplu: dacă **n=12** și **v=(21, 412, 2, 2026, 208, 523, 7, 128, 3174, 2026, 46, 8)** se va returna valoarea **4** (numerele "călduroase" din șir sunt **412, 2, 2026, 208, 523, 7, 2026, 46, 8**. Cea mai mare valoare este 2026. Prima ei apariție este pe poziția 4)

2. (10p.) Scrieți un program C/C++ care citește de la tastatură un text de maximum 500 de caractere, format doar din litere ale alfabetului englez și spații, în care cuvintele sunt separate prin câte un singur spațiu.

Asupra textului se va aplica următoarea prelucrare: pentru fiecare lungime posibilă a cuvintelor ($lu = 1, 2, 3, \dots$, până la lungimea maximă a unui cuvânt din text), primul cuvânt de lungime lu se va interschimba cu ultimul cuvânt de aceeași lungime lu , al doilea cuvânt de lungime lu cu penultimul cuvânt de lungime lu și așa mai departe, dacă astfel de cuvinte există.

Programul va afișa pe ecran textul obținut după toate aceste interschimbări.

Exemplu: dacă se citește textul **Ana a mers azi cu bicicleta patru kilometri spre mare**, se va afișa **azi a mare Ana cu kilometri patru bicicleta spre mers**

Dacă se citește textul **Mihai are trei pisici**, se va afișa **Mihai are trei pisici**

3. Se dă fișierul **bac.txt** care conține pe prima linie două numere naturale **n** ($2 \leq n \leq 100000$) și **p** ($2 \leq p \leq 1000$) și pe a doua linie un șir cu **n** numere naturale din intervalul $[1; 30000]$. Se cere să se scrie pe ecran numărul de perechi distincte de elemente de pe poziții diferite în șirul dat a căror sumă este multiplu de **p**. Proiectați un algoritm eficient din punct de vedere al timpului de executare.

Exemplu: pentru **n=5, p=3** și șirul **9 2 4 7 3** se scrie pe ecran **3** deoarece perechile de elemente cu suma multiplu de 3 sunt **(9, 3)**, **(2, 4)** și **(2, 7)**; pentru **n=4, p=6** și șirul **1 1 6 4** se scrie pe ecran **0** deoarece nu există perechi de elemente cu suma multiplu de 6; pentru **n=8, p=4** și șirul **9 22 3 2 6 4 8 14** se scrie pe ecran **8** deoarece perechile de elemente cu suma multiplu de 4 sunt **(9, 3)**, **(22, 2)**, **(22, 6)**, **(22, 14)**, **(2, 6)**, **(2, 14)**, **(6, 14)** și **(4, 8)**.

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)