

Examenul național de bacalaureat 2024

**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 muchie are extremități distincte și oricare două 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.

- Determinați valoarea maximă posibilă a expresiei alăturate, unde x reprezintă un număr întreg. $(x \% 10 + 5) \% 3 * 4$
a. 7 b. 8 c. 9 d. 10
- Subprogramele **f** și **g** sunt definite alăturat. Indicați rezultatul afișării pentru apelul **g(5)**.

| | |
|---|--|
| <pre>int f(int n) { if(n==0) return 0; if(n%2==0) return n+f(n/2); return f(n/2); }</pre> | <pre>void g(int n) { for(int i=1;i<=n;i++) cout<<f(i)<<" "; printf("%d ",f(i)); }</pre> |
|---|--|

a. 1 1 4 1 6 b. 0 0 0 0 0 c. 0 2 0 2 2 d. 0 2 0 6 2
- Utilizând metoda backtracking se generează toate numerele de **k** cifre cu cifre strict crescătoare și cu parități diferite pe poziții consecutive. Spre exemplu, pentru **k=3** primele 5 soluții sunt 123, 125, 127, 129, 145. Utilizând același algoritm pentru **k=5**, determinați al 3-lea număr generat în ordine descrescătoare.
a. 34579 b. 56789 c. 36789 d. 45678
- Variabila **m** este declarată alăturat. Indicați o expresie corectă din punct de vedere sintactic.

```
struct magazin
{ char denumire[25];
  struct produs
  { char denumire[25];
    int stoc;
  }p[10];
}m;
```


a. `strcpy(m.p[0].denumire, m.denumire + m.p[0].stoc);`
b. `strcpy(m.p[0].denumire[0], m.p[0].stoc);`
c. `strcat(m.p[0].denumire, m.p[0].stoc);`
d. `strcmp(m, p[0]);`
- Se consideră un graf neorientat cu 67 noduri și 2024 muchii. Câte muchii trebuie eliminate pentru ca graful să devină arbore?
a. 1958 b. 1955 c. 1942 d. 187

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 citesc valorile 1234 5623 1111 1034 0. **(6p.)**
- b. Scrieți două seturi de date distincte astfel încât, în urma executării algoritmului să se afișeze valoarea 3210. **(6p.)**

```
s ← 0
citește nr (număr natural de 4 cifre)
cât timp nr ≠ 0 execută
    dacă nr % 2 = 0 atunci
        nr ← nr / 100 * 100 + nr % 10 * 10 + nr / 10 % 10
    altfel
        nr ← nr / 100 % 10 * 1000 + nr / 1000 * 100 + nr % 100
    s ← s + nr
    citește nr
scrie s
```

- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura **cât timp...** **execută** cu o structură repetitivă cu test final. **(6p.)**

2. Un graf orientat cu 6 noduri, numerotate de la 1 la 6, este reprezentat prin liste de adiacență alăturate. Pentru acest graf scrieți matricea de adiacență. **(6p.)**

| | |
|------------|------|
| 1: 2, 3, 5 | 4: 6 |
| 2: 4 | 5: - |
| 3: - | 6: 4 |

3. Se consideră un teren ce este acoperit cu pătrate de iarbă și pavaj. Acest teren este memorat într-un tablou bidimensional M , format din 3 linii și 7 coloane, unde iarba este reprezentată cu valoarea 0, iar pavajul cu 1. Pe o singură coloană se poate afla maxim un pavaj. Scrieți o secvență de instrucțiuni astfel încât aceasta să afișeze pe ecran valoarea 1 dacă există un drum sau 0 în caz contrar. Un drum există dacă pornește din prima până la ultima coloană, fiind continuat pe același rând sau pe diagonală. **Exemplu:** pentru drumul alăturat se afișează valoarea 1. **(6p.)**

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |

SUBIECTUL al III-lea

(30 de puncte)

1. Un număr este considerat perfect dacă acesta este prim, iar suma cifrelor sale este pară. Subprogramul **perfect** are doi parametri, n prin care primește un număr natural din intervalul $[10^2, 10^9]$ și k , o cifră. Subprogramul afișează pe ecran, în ordine descrescătoare, primele k numere perfecte mai mici decât n sau mesajul **nu exista** în cazul în care nu există suficiente numere perfecte. **Exemplu:** dacă $n=100$ și $k=3$ se va afișa 97 79 73. **(10p.)**

2. Două cuvinte rimează perfect dacă ambele au același număr de silabe, iar ultimele 2 litere sunt identice. Spre exemplu, **soare** și **mare** sunt rime perfecte, însă **are** și **ardoare** sunt doar rime simple. Scrieți un program C/C++ care citește de la tastatură un text cu cel mult 100 de caractere ce conține cuvinte formate din litere mici ale alfabetului englez, separate prin câte un spațiu. Cuvintele sunt despărțite în silabe prin simbolul '-' și nu pot exista cuvinte precum **spă-lă-to-ri-e**, care au o singură literă drept ultimă silabă. Programul determină numărul de perechi de cuvinte care rimează perfect sau mesajul **nu exista** în caz contrar.

Exemplu: Pentru textul **mai-ne mer-gem la ma-re cu o floa-re si sper sa stam in soa-re sa fa-cem pla-ja** se afișează pe ecran 5. (Perechile sunt: **mergem-facem**, **mare floare**, **mare-soare**, **floare-soare**, **sa-sa**). **(10p.)**

3. Șirul de mai jos este definit alăturat:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, ...

$$f_n = \begin{cases} 1, & \text{pentru } n = 1 \\ 2, & \text{pentru } n = 2 \\ 2 \cdot f_{n-2} + f_{n-1}, & \text{pentru } n \geq 3 \end{cases}$$

Se citește de la tastatură un număr natural n ($n \in [3, 50]$) și se cere să se scrie în fișierul **bac.out**, separate prin câte un spațiu, primele n elemente din șir, în ordine descrescătoare. Proiectați un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate.

Exemplu: dacă $n=7$ atunci fișierul conține numerele: 64 32 16 8 4 2 1

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