

Examenul de bacalaureat național 2019
Proba E. d)
Informatică
Limbajul C/C++

MODEL

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

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 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. Variabila întregă **n** memorează un număr natural. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în **n** este divizibil cu 20, dar **NU** și cu 19.

- | | |
|----------------------------------------------------|-----------------------------------------------|
| a. <code>n%380==0 && n/20==0</code> | b. <code>n%380!=0 !(n%19==0)</code> |
| c. <code>n%20==0 && n/19==0</code> | d. <code>!(n%20!=0 n%19==0)</code> |

2. Subprogramul **f** este definit alăturat. Indicați apelul care determină afișarea, în ordine strict descrescătoare, a tuturor divizorilor proprii pozitivi ai numărului 1000 (divizori diferiți de 1 și de 1000).

```
void f (int n, int d)
{ if(d<n/2) f(n, d+1);
  if(n%d==0)
    cout<<d<<' '; | printf("%d ",d);
}
```

- | | | | |
|-----------------------------------|----------------------------------|----------------------------------|---------------------------------|
| a. <code>f(1000,2);</code> | b. <code>f(999,2);</code> | c. <code>f(500,2);</code> | d. <code>f(32,2);</code> |
|-----------------------------------|----------------------------------|----------------------------------|---------------------------------|

3. Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte 3 esențe distincte din mulțimea {agar, geranium, iasomie, paciuli, tuberoze}. Primele patru soluții obținute sunt, în această ordine: (agar, geranium, iasomie), (agar, geranium, paciuli), (agar, geranium, tuberoze) și (agar, iasomie, paciuli). Indicați soluția generată imediat înainte de (geranium, iasomie, paciuli).

- | | |
|---------------------------------|------------------------------|
| a. (agar, iasomie, paciuli) | b. (agar, paciuli, tuberoze) |
| c. (geranium, paciuli, iasomie) | d. (geranium, agar, iasomie) |

4. Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați” (6,5,7,5,9,9,6,7,0,5). Numărul nodurilor de tip „frunză” ale arborelui este:

- | | | | |
|------|------|------|------|
| a. 4 | b. 5 | c. 6 | d. 7 |
|------|------|------|------|

5. Un graf neorientat are 10 muchii și este conex. Numărul maxim de noduri ale sale este:

- | | | | |
|------|------|-------|-------|
| a. 8 | b. 9 | c. 10 | d. 11 |
|------|------|-------|-------|

SUBIECTUL al II-lea

(40 de puncte)

1. Se consideră algoritmul alăturat, reprezentat în pseudocod.

- a) Scrieți valorile afișate dacă se citește numărul 7. (6p.)
- b) Scrieți cel mai mic și cel mai mare număr care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, ultima valoare afișată să fie 10. (6p.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind una dintre structurile **cât timp...execută** cu o structură repetitivă de alt tip. (6p.)

```
citește n
(număr natural)
k ← 1
cât timp n ≥ 1 execută
| dacă n > k atunci i ← k
| altfel i ← n
| ■
| n ← n - i
| cât timp i ≥ 1 execută
| | scrie k, ' '; i ← i - 1
| | ■
| k ← k + 1
| ■
```

2. Pentru un număr complex se memorează următoarele date: partea reală și partea imaginară (numere reale). Variabila **z** memorează simultan date pentru fiecare dintre cele 20 de numere complexe. Știind că expresia C/C++ de mai jos are valoarea sumei dintre partea reală și partea imaginară ale primului număr complex dintre cele precizate, scrieți definiția unei structuri cu eticheta **complex** care să permită memorarea datelor unui număr complex, și declarați corespunzător variabila **z**.

`z[0].pre+z[0].pim`

(6p.)

3. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, având inițial toate elementele nule. Fără a utiliza alte variabile decât cele menționate, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **a** să memoreze tabloul alăturat.

`for(i=1;i<=5;i++)
 for(j=1;j<=5;j++)`

.....

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul **CifrePrime** are un singur parametru, **n**, prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează suma cifrelor prime ale lui **n**.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=1235405**, atunci subprogramul returnează 15, iar dacă **n=140**, atunci subprogramul returnează 0.

(10p.)

2. Într-un text cu cel mult 100 de caractere, cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe spații.

Scrieți un program C/C++ care citește de la tastatură un astfel de text, cu cel puțin trei cuvinte, și construiește în memorie un șir de caractere format din prima consoană a primului cuvânt, urmată de prima vocală a celui de al doilea cuvânt, respectiv de ultima literă a ultimului cuvânt, în ordinea în care acestea apar în text. Șirul obținut se afișează pe ecran, iar dacă nu se poate obține un astfel de șir, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele a, e, i, o, u.

Exemplu: pentru textul el prefera sa mearga la schi

se afișează pe ecran șirul lei

iar pentru textul ei prefera sa mearga la schi

se afișează pe ecran mesajul **nu exista**

(10p.)

3. Un interval este numit **prieten de grad n** al unui șir dacă sunt exact **n** termeni ai șirului cu valori din interval și dacă toate numerele întregi care aparțin intervalului sunt valori ale unor termeni ai șirului. Fișierul **bac.txt** conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 10^2]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul maxim **n** cu proprietatea că există un interval prieten de grad **n** al șirului aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul conține numerele

10 10 11 3 4 2 49 4 2 3 21 2 27 12 13 14 15 5

atunci se afișează pe ecran 8 (intervalului $[2, 5]$ îi aparțin 8 termeni ai șirului)

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

(2p.)

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

(8p.)