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

MODEL

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

- **Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.**
- **Timpul de lucru efectiv este de 3 ore.**
- **Identificatorii utilizaţi în rezolvări trebuie să respecte precizările din enunţ (bold), iar în lipsa unor precizări explicite, notaţiile trebuie să corespundă cu semnificaţiile 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 **(40 de puncte)**

Pentru fiecare dintre itemii de la 1 la 10, scrieţi pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila întreagă **n** memorează un număr natural. Indicați expresia C/C++ care are valoarea **1** dacă şi numai dacă numărul memorat în **n** este divizibil cu **20**, dar **NU** şi cu **19**.

a. `n%380==0 && n/20==0` **b.** `n%380!=0 || !(n%19==0)`
c. `n%20==0 && n/19==0` **d.** `!(n%20!=0 || n%19==0)`

2. În secvențele de mai jos, reprezentate în pseudocod, valoarea inițială a variabilei **x** este **20**.

```
┌repetă
│ scrie 19; x←x-1
└până când ........
```

```
┌cât timp x≥1 execută
│ x←x-1; scrie 19
└■
```

Cele două secvențe sunt echivalente dacă se înlocuiesc punctele de suspensie cu expresia:

a. `x<1` **b.** `x>1` **c.** `x≤1` **d.** `x≥1`

3. În secvența alăturată, variabilele **i** şi **j** sunt de tip întreg.

```
for(i=1;i<=4;i++)
  for(j=1;j<=4;j++)
    cout<<.......<<' '; | printf("%d ",.......);
```

Indicați expresia care poate înlocui punctele de suspensie, astfel încât în urma executării secvenţei obţinute să se afișeze pe ecran, strict crescător, toate numerele naturale din intervalul **[1,16]**.

a. `i+(j-1)*4` **b.** `i+(j+1)*4` **c.** `(i-1)*4+j` **d.** `(i+1)*4+j`

4. Variabila **m** memorează un tablou bidimensional cu **10** linii și **10** coloane, numerotate de la **0** la **9**. Indicați expresia care are valoarea elementului aflat pe ultima linie și ultima coloană.

a. `m(9,9)` **b.** `m(9)(9)` **c.** `m[9,9]` **d.** `m[9][9]`

5. Variabila **z** memorează partea reală şi partea imaginară ale unui număr complex. Indicați o declarare posibilă a variabilei **z**, știind că expresia **z.pre+z.pim** are valoarea sumei dintre partea reală și partea imaginară ale numărului complex corespunzător acestei variabile.

a. `struct z {float pre,pim;};` **b.** `struct {float pre, pim;} z;`
c. `{float pre, pim} struct z;` **d.** `z {float pre, pim} struct;`

6. Subprogramul **f** este definit alăturat. Indicați apelul care determină afişarea, 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. `f(500,1);` **b.** `f(1000,2);` **c.** `f(1,1000);` **d.** `f(2,999);`

7. Interclasând descrescător tablourile unidimensionale **(1,5,7,10,20)** și **(19,10,9,8,5,1)** se obține:

a. `(20,19,10,9,8,5,1)` **b.** `(20,19,10,10,9,8,7,5,5,1,1)`
c. `(20,19,10,9,8,7,5,1)` **d.** `(20,19,10,10,9,7,8,5,5,1,1)`

8. Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte **3** esențe distincte din mulţimea {**agar**, **geranium**, **iasomie**, **paciuli**, **tuberoze**}. Primele patru soluţii obţinute sunt, în această ordine: (**agar**, **geranium**, **iasomie**), (**agar**, **geranium**, **paciuli**), (**agar**, **geranium**, **tuberoze**) şi (**agar**, **iasomie**, **paciuli**). Indicaţi soluția generată imediat înainte de (**geranium**, **iasomie**, **paciuli**).

a. (**agar**, **iasomie**, **paciuli**) **b.** (**agar**, **paciuli**, **tuberoze**)

c. (**geranium**, **paciuli**, **iasomie**) **d.** (**geranium**, **agar**, **iasomie**)

9. Un arbore cu **10** noduri, numerotate de la **1** la **10**, este reprezentat prin vectorul de „taţi” **(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**

10. 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 (20 de puncte)

Algoritmul alăturat este reprezentat în pseudocod.

1. Scrieţi valorile afişate dacă se citește numărul **7**. **(6p.)**

2. Scrieţi cel mai mic și cel mai mare număr care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, ultima valoare afişată să fie **10**. **(4p.)**

3. Scrieţi programul C/C++ corespunzător algoritmului dat. **(10p.)**

```
citeşte n (număr natural nenul)
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
└■
```

SUBIECTUL al III-lea (30 de puncte)

1. Subprogramul **CifrePrime** are un singur parametru, **n**, prin care primeşte un număr natural ($n \in [0, 10^9]$). Subprogramul returnează suma cifrelor prime ale lui **n**.
Scrieţi definiţia completă a subprogramului.
Exemplu: dacă **n=1235405**, atunci subprogramul returnează valoarea **15**. **(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.)**