

**Examenul de bacalaureat național 2017**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

**Varianta 7**

**Filiera teoretică, profilul real, specializările: matematică-informatică**  
**matematică-informatică intensiv informatică**  
**Filiera vocațională, profilul militar, specializarea matematică-informatică**

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- În rezolvările cerute, identificatorii utilizați 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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

**SUBIECTUL I** (30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Expresia C/C++ alăturată are valoarea: (4p.) | 3+7/4+3
- a. 1                      b. 5.5                      c. 7                      d. 7.75

**2. 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 ce se afișează dacă se citesc, în această ordine, numerele **4, 5, 24, 3, 45**. (6p.)

- b) Dacă primul număr citit este **3**, scrieți un set de numere distincte care pot fi citite în continuare, astfel încât, în urma executării algoritmului, să se afișeze valoarea **1**. (4p.)

- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat prima structură **cât timp...execută** cu o structură de tip **pentru...execută**. (6p.)

```
citește n  
  (număr natural nenul)  
k ← 0  
i ← 1  
cât timp i ≤ n execută  
  citește x  
    (număr natural nenul)  
  y ← 2  
  cât timp x > 1 și x%y ≠ 0 execută  
    y ← y + 1  
  ■  
  dacă k < [x/y] atunci  
    k ← [x/y]  
  ■  
  i ← i + 1  
  ■  
scrie k
```

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

**SUBIECTUL al II-lea**

**(30 de puncte)**

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

- În declararea alăturată, în câmpurile **s** și **d** ale structurii sunt memorate extremitatea stângă, respectiv dreaptă ale fiecăruia dintre cele 20 de intervale închise. O expresie C/C++ a cărei valoare este extremitatea stângă a primului interval este: (4p.)  

```
struct interval
{ float s,d;
}v[20];
```

  - v[0].s**
  - s.v[0]**
  - v(0).s**
  - s(0).v**
- Într-un arbore cu 50 de noduri, numerotate de la 1 la 50, rădăcina este nodul 1, iar tatăl oricărui alt nod **i** al său este nodul numerotat cu  $\lfloor i/2 \rfloor$ . Lungimea lanțului cu o extremitate în nodul 14 și cealaltă extremitate în nodul 47 este: (4p.)  
  - 5
  - 8
  - 16
  - 33

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

- Un graf neorientat cu 4 noduri, numerotate de la 1 la 4, este reprezentat prin matricea de adiacență alăturată. Scrieți mulțimea nodurilor și mulțimea muchiilor unui subgraf al acestui graf, diferit de el însuși, în care cel puțin un nod să aibă gradul 1 și cel puțin un nod să aibă gradul 2. (6p.)  

0	1	0	1
1	0	0	1
0	0	0	1
1	1	1	0
- În secvența alăturată variabilele **n** și **i** sunt de tip întreg, iar variabila **s** permite memorarea unui cuvânt, șir de cel mult 20 de caractere. Cuvintele citite sunt formate din litere mici ale alfabetului englez și sunt separate prin Enter.  

```
n=.....;
for(i=1;i<=11;i++)
{ cin>>s; | scanf("%s",s);
  .....
}
```

Fără a utiliza alte variabile, scrieți secvența înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **n** să memoreze numărul de cuvinte citite pentru care subșirul format din ultimele trei caractere ale lor coincide cu șirul **are**, iar acesta **NU** mai apare pe alte poziții în cuvânt.  
**Exemplu:** dacă se citesc cuvintele alăturate, variabila **n** are valoarea 3. (6p.)

la  
proiectare  
se  
are  
in  
vedere  
evitarea  
deformatiilor  
areolare  
si  
liniare

- Scrieți un program C/C++ care citește de la tastatură un număr natural par **n** ( $n \in [2, 50]$ ) și elementele unui tablou bidimensional cu **n** linii și **n** coloane, numere reale, apoi transformă tabloul în memorie, inserând o linie nouă, la mijlocul său, valoarea fiecărui element al acesteia fiind egală cu media aritmetică a elementelor aflate pe coloana corespunzătoare lui în tabloul citit. Tabloul obținut se afișează pe ecran, câte o linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru **n=4** și tabloul

1.5	0.75	3	1.2	se obține tabloul:	1.5	0.75	3	1.2
2.75	1.25	2.25	1.5		2.75	1.25	2.25	1.5
4.5	1.15	3	4.5		2.25	1	2.5	2.55
0.25	0.85	1.75	3		4.5	1.15	3	4.5
					0.25	0.85	1.75	3

(10p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Subprogramul **f** este definit alăturat.  
Valoarea lui **f(1234)** este:

(4p.)

```
int f(int n)
{ if (n!=0)
  return (n%2)*(n%10)+f(n/10);
  return 0;
}
```

- a. 0                                      b. 4                                      c. 6                                      d. 10

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Fiind date o mulțime finită și un set de perechi de forma  $(i, j)$ , unde  $i$  și  $j$  sunt elemente ale mulțimii, se numește **șir sortat topologic asociat mulțimii** un șir format din toate elementele mulțimii, astfel încât pentru orice pereche  $(i, j)$  dintre cele date, elementul  $i$  să se găsească înaintea lui  $j$  în șir.  
Utilizând metoda backtracking, se generează toate șirurile sortate topologic ale mulțimii  $\{1, 2, 3, 4, 5, 6\}$ , având în vedere perechile  $(4, 6)$ ,  $(3, 4)$ ,  $(3, 1)$ ,  $(2, 5)$ . Primele trei soluții generate sunt, în această ordine:  $(2, 3, 1, 4, 5, 6)$ ,  $(2, 3, 1, 4, 6, 5)$ ,  $(2, 3, 1, 5, 4, 6)$ . Scrieți cea de a patra și cea de a cincea soluție, în ordinea generării acestora. (6p.)
3. Subprogramul **triplete** are trei parametri, **a**, **b** și **c**, prin care primește câte un număr natural din intervalul  $[0, 10^4]$  ( $a < b$ ). Subprogramul afișează pe ecran toate tripletele de numere naturale, **x**, **y** și **z**, soluții ale ecuației  $x+y+z=c$ , cu proprietatea că  $a \leq x \leq y \leq z \leq b$ . Fiecare triplet afișat este încadrat între acolade, iar numerele **x**, **y** și **z** sunt afișate în această ordine, separate prin câte o virgulă. Dacă nu există niciun triplet cu proprietatea cerută, se afișează mesajul **nu exista**.  
Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă **a=2**, **b=4** și **c=8**, se afișează pe ecran tripletele  $\{2, 2, 4\}$   $\{2, 3, 3\}$   
iar dacă **a=5**, **b=8**, **c=8** se afișează pe ecran mesajul **nu exista**. (10p.)
4. Un șir de numere este o **progresie aritmetică de rație  $r$**  dacă oricare termen al său, cu excepția primului, se obține din cel care îl precede, prin adunarea la acesta a numărului  $r$ .  
**Exemplu:** șirul 12, 14, 16, 18, 20 este o progresie de rație 2.  
Fișierul **bac.in** conține un șir de cel mult  $10^6$  numere naturale din intervalul  $[0, 10^3]$ , separate prin câte un spațiu.  
Se cere să se verifice dacă există un număr natural  $r$ , astfel încât toate numerele **distincte** din șir să poată fi rearanjate, pentru a forma o progresie aritmetică de rație  $r$ . Se afișează pe ecran numărul  $r$ , sau mesajul **NU**, dacă nu există un astfel de număr. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.  
**Exemplu:** dacă fișierul conține numerele  
180 30 80 280 130 330 230 30 30 330 80  
se afișează pe ecran  
50  
a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b) Scrieți programul C/C++ corespunzător algoritmului descris. (8p.)