

Examenul de bacalaureat național 2013
Proba E. d)
Informatică
Limbajul C/C++

Varianta 6

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. Valoarea expresiei **C/C++** alăturate este: (4p.) | 7+5/2
- a. 6 b. 9 c. 9.5 d. 10

2. Se consideră algoritmul alăturat, reprezentat în pseudocod.

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

- a) Scrieți numărul afișat dacă pentru variabila **a** se citește valoarea **65**, iar pentru variabila **b** se citește valoarea **80**. (6p.)
- b) Dacă pentru variabila **a** se citește valoarea **1234**, scrieți cel mai mare număr de patru cifre care poate fi citit pentru variabila **b** astfel încât, în urma executării algoritmului, valoarea afișată să fie **5**. (4p.)

```
citește a,b  
    (numere naturale nenule,  $a \leq b$ )  
nr ← 0  
pentru  $i \leftarrow a, b$  execută  
    |  $x \leftarrow i$   
    |  $c \leftarrow x \% 10$   
    | cât timp  $x \neq 0$  și  $x \% 10 = c$  execută  
    |   |  $x \leftarrow [x/10]$   
    |   └─┐  
    └─┐  
    | dacă  $x = 0$  atunci  
    |   | nr ← nr + 1  
    |   └─┐  
    └─┐  
    scrie nr
```

- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **pentru...execută** cu o structură repetitivă cu test final. (6p.)
- 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.

1. Variabila **i** este de tip întreg ($0 \leq i \leq 99$), iar variabila **a** memorează elementele unui tablou bidimensional cu 100 de linii și 100 de coloane, numerotate de la 0 la 99. În limbajul C/C++, un element aflat pe linia **i** și pe diagonala principală a tabloului poate fi accesat prin: **(4p.)**
- a. **a[i, i]** b. **a(i, i)** c. **a(i)(i)** d. **a[i][i]**
2. Se consideră un graf neorientat conex cu 50 de noduri și 52 de muchii. Numărul minim de muchii ce pot fi eliminate astfel încât graful parțial obținut să nu aibă niciun ciclu este: **(4p.)**
- a. **1** b. **2** c. **3** d. **4**

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

3. Se consideră arborele cu 8 noduri, numerotate de la 1 la 8, reprezentat prin vectorul de „tați”: (3, 0, 2, 5, 2, 5, 2, 5). Enumerați nodurile arborelui care au același „tată” cu nodul 4. **(6p.)**

4. În declarațiile alăturate:
- variabila **p** memorează coordonatele unui punct în sistemul de coordonate **xOy**;
 - variabila **d** memorează în câmpurile **A** și **B** coordonatele, în același sistem de coordonate, ale vârfurilor din stânga – sus, respectiv din dreapta – jos ale unui dreptunghi cu laturile paralele cu axele sistemului de coordonate.
- ```
struct punct {
 int x,y;
} p;

struct dreptunghi {
 punct A, B;
} d;
```

Scrieți o expresie C/C++ care să aibă valoarea 1 dacă punctul corespunzător variabilei **p** se află în interiorul dreptunghiului corespunzător variabilei **d** (dar nu pe laturile acestuia) sau valoarea 0 în caz contrar. **(6p.)**

5. Se consideră un text cu cel mult 100 de caractere (litere mici ale alfabetului englez și spații), în care cuvintele sunt separate prin unul sau mai multe spații. Înaintea primului cuvânt și după ultimul cuvânt nu există spații. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și determină transformarea acestuia în memorie prin eliminarea unor spații, astfel încât între oricare două cuvinte alăturate să rămână exact un spațiu. Programul afișează pe ecran textul obținut.

**Exemplu:** pentru textul

**in vacanta    plec la                      mare**  
se obține și se afișează  
**in vacanta plec la mare**

**(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. Se consideră subprogramul **f**, definit alăturat. Indicați ce se afișează în urma apelului de mai jos. **f(3);** (4p.)
- ```
void f (int n)
{ if (n!=0)
  { f (n-1);
    cout<<n; | printf("%d",n);
  }
}
```
- a. 12 b. 123 c. 321 d. 3210

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

2. Utilizând metoda backtracking, se generează în ordine lexicografică toate șirurile de câte 5 litere distincte din mulțimea {**A, B, C, D, E**}, astfel încât în fiecare șir litera **D** precede literele **A** și **B**. Primele cinci soluții generate sunt, în această ordine: **CDABE, CDAEB, CDBAE, CDBEA, CDEAB**. Scrieți cea de a șasea și cea de a șaptea soluție, în ordinea generării acestora. (6p.)
3. Se consideră subprogramul **sub**, cu trei parametri:
- **n**, prin care primește un număr natural ($2 < n < 50$);
 - **v**, prin care primește un tablou unidimensional cu **n** elemente, numere naturale cu cel mult 4 cifre;
 - **k**, prin care primește un număr natural ($1 < k \leq n$).
- Subprogramul returnează suma primelor **k** elemente cu valoare impară ale tabloului. Dacă nu există **k** elemente impare în tablou, subprogramul returnează valoarea -1. Scrieți definiția completă a subprogramului.
- Exemplu:** dacă **n=8**, **v=(2, 7, 6, 8, 3, 7, 5, 1)**, **k=3**, atunci subprogramul returnează valoarea 17 ($7+3+7=17$). (10p.)
4. Fiind date două numere **a** și **b**, îl numim pe **a** sufix al lui **b** dacă **a** este egal cu **b** sau dacă **b** se poate obține din **a** prin alipirea la stânga a unor noi cifre.
- Exemplu:** 12 este sufix al lui 12, iar 15 este sufix al lui 31415.
- Fișierul **bac.txt** conține pe prima linie un număr natural **x**, cu cel mult nouă cifre, iar pe a doua linie un șir de cel puțin două și cel mult 1000000 de numere naturale cu cel mult nouă cifre. Numerele din șir sunt separate prin câte un spațiu.
- Se cere să se afișeze pe ecran ultimul termen al șirului care are ca sufix numărul **x**. Dacă în șir nu există o astfel de valoare, pe ecran se afișează mesajul **Nu exista**.
- Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.
- Exemplu:** dacă fișierul **bac.txt** conține numerele
- 12
3445 89312 1245 12 67120 312 1234578
- atunci pe ecran se afișează 312.
- a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. (4p.)
- b) Scrieți programul **C/C++** corespunzător algoritmului descris. (6p.)