

**Examenul de bacalaureat național 2014**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

**Varianta 2**

**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. Variabilele **x**, **y** și **z** sunt de tip întreg și memorează câte un număr natural nenul. Dacă expresia **C/C++** alăturată are valoarea 1, indicați șirul crescător format cu valorile acestor variabile, în ordinea precizată mai jos. (4p.)

- a. **x, y, z**                      b. **y, z, x**                      c. **z, x, y**                      d. **z, y, x**

**2. Se consideră algoritmul alăturat, reprezentat în pseudocod.**

S-a notat cu **x%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 se citește valoarea 162453. (6p.)
- b) Scrieți două numere de patru cifre distincte care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze valoarea 0. (4p.)
- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă cu test final. (6p.)
- d) Scrieți programul **C/C++** corespunzător algoritmului dat. (10p.)

```
citește n
(n număr natural nenul)
m ← n
p ← 1
cât timp n ≥ p * 10 execută
    c1 ← [n/p] % 10
    c2 ← [n/(p*10)] % 10
    dacă c1 > c2 atunci
        n ← n - (c1 - c2) * p
        p ← p * 10
        n ← n + (c1 - c2) * p
    altfel
        p ← p * 10
    sfârșit
dacă n = m atunci
    scrie 0
altfel
    scrie n
sfârșit
```

**SUBIECTUL al II-lea**

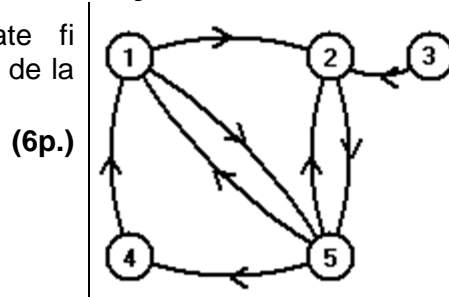
**(30 de puncte)**

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

1. Pentru a putea memora un tablou bidimensional cu maximum 21 de elemente, numere reale, variabila **A** se poate declara astfel: **(4p.)**
  - a. `A[3..7] int;`
  - b. `A[3][7] float;`
  - c. `int A[3;7];`
  - d. `float A[3][7];`
2. Se consideră un graf neorientat conex și fără cicluri, în care gradul oricărui nod este mai mic sau egal cu 4. Dacă șase dintre nodurile sale au gradul egal cu 1, atunci numărul maxim de noduri cu gradul egal cu 4 este: **(4p.)**
  - a. 0
  - b. 1
  - c. 2
  - d. 3

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

3. Scrieți matricea de adiacență prin care poate fi reprezentat graful orientat cu 5 vârfuri, numerotate de la 1 la 5, ilustrat în figura alăturată. **(6p.)**



4. Se consideră următoarea operație prin care se transformă un arbore cu rădăcină: se elimină din arbore nodul „frunză” numerotat cu valoarea minimă, marcându-se nodul „tată” al acestuia. Scrieți vectorul de „tați” al unui arbore cu 6 noduri, numerotate de la 1 la 6, în care nodul 1 este rădăcină, știind că dacă asupra acestui arbore se efectuează de patru ori succesiv operația de transformare menționată mai sus, se marchează, în această ordine, nodurile 5, 1, 1, 1. **(6p.)**
5. Se consideră un text cu cel mult 100 de caractere, în care cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și determină transformarea acestuia în memorie, astfel încât din fiecare cuvânt format dintr-un număr impar de litere (cel puțin trei) să se elimine litera aflată pe poziția din mijloc, ca în exemplu. Programul afișează pe ecran textul obținut, iar dacă nu există niciun cuvânt modificat, afișează pe ecran mesajul **nu exista**.  
**Exemplu:** pentru textul  
`pictura prin aceea arata o pace profunda`  
se afișează  
`picura prin acea arta o pace profunda` **(10p)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. Utilizând metoda backtracking, se generează toate posibilitățile de a obține suma 4 cu numere naturale nenule. Două sume sunt distincte dacă diferă prin cel puțin un termen. Soluțiile generate sunt, în această ordine, 1+1+1+1, 1+1+2, 1+3, 2+2. Aplicând același algoritm pentru a genera toate posibilitățile de a obține suma 6, dacă prima soluție generată este 1+1+1+1+1+1, atunci soluția 1+2+3 este generată: **(4p.)**
- a. a 6-a                      b. a 7-a                      c. a 8-a                      d. a 9-a

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

2. Se consideră subprogramul **f**, definit alături. Scrieți valorile **f(5,5)** și **f(10,21)**. **(6p.)**
- ```
int f(int a, int b)
{
    if (a==b) return 0;
    if (b/a==0) return a+b;
    return f(a+2,b-3);
}
```
3. Un număr natural nenul se numește **perfect** dacă este egal cu suma divizorilor săi naturali strict mai mici decât el.  
**Exemplu:** 28 este număr perfect pentru că  $28=1+2+4+7+14$ .  
Se consideră subprogramul **perfect**, cu doi parametri, **a** și **b**, prin care primește câte un număr natural ( $2 \leq a < b \leq 10^9$ ). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele perfecte din intervalul **[a,b]**. Dacă în interval nu există astfel de numere, subprogramul afișează pe ecran mesajul **nu exista**. Scrieți definiția completă a subprogramului.  
**Exemplu:** pentru **a=5** și **b=30**, se afișează pe ecran: 28 6 **(10p.)**
4. Numim **secvență uniformă** a unui șir de numere naturale un subșir al acestuia, format din termeni cu aceeași valoare, aflați pe poziții consecutive în șirul dat. Lungimea secvenței este egală cu numărul de termeni ai acesteia.  
Fișierul **bac.txt** conține un șir de cel puțin două și cel mult 1000000000 de numere naturale din intervalul **[0, 10<sup>9</sup>]**. Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni egali pe poziții consecutive.  
Se cere să se determine o secvență uniformă de lungime maximă în șirul aflat în fișier și să se afișeze pe ecran lungimea acestei secvențe și, pe o linie nouă, separați prin câte un spațiu, termenii acesteia. Dacă sunt mai multe astfel de secvențe, se afișează doar termenii ultimei dintre acestea. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei necesare și al timpului de executare.  
**Exemplu:** dacă fișierul **bac.txt** conține numerele  
2 3 3 3 3 5 4 4 11 11 11 11 16 11 11 11 11 15 15  
atunci pe ecran se afișează valorile  
4  
11 11 11 11  
a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**  
b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**