

Subiectul III (30 de puncte) - Varianta 029

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

1. În câte dintre permutările elementelor mulțimii $\{ 'I', 'N', 'F', 'O' \}$ vocalele apar pe poziții consecutive? (4p.)
- a. 4 b. 6 c. 24 d. 12

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

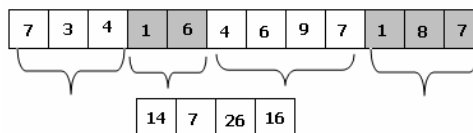
2. Ce se afișează ca urmare a apelului `p(123)`; dacă subprogramul `p` are definiția alăturată? (6p.)
- ```
void p (int x)
{ if(x!=0){p(x/10);
 cout<<x%10; | printf("%d",x%10);
}
}
```

3. Scrieți programul `c/c++` care citește trei numere naturale  $a, b, c$  ( $a \leq b$ ) din intervalul  $[1, 10000]$  și afișează pe ecran numărul multiplilor lui  $c$  din intervalul  $[a, b]$ .

**Exemplu:** pentru  $a=10, b=27, c=5$  se afișează valoarea 4. (10p.)

4. Se consideră două tablouri unidimensionale **A** și **B** cu elemente numere naturale din intervalul  $[1, 10000]$ . Spunem că tabloul **A** "se poate reduce" la tabloul **B** dacă există o împărțire pe secvențe de elemente aflate pe poziții consecutive în tabloul **A** astfel încât prin înlocuirea secvențelor cu suma elementelor acestora să se obțină, în ordine, elementele tabloului **B**.

De exemplu tabloul



se poate reduce la tabloul

- a) Scrieți definiția completă a subprogramului **suma**, cu trei parametri, care:
- primește prin parametri:
  - **A** un tablou unidimensional cu maximum 100 elemente numere naturale din intervalul  $[1, 10000]$
  - $i$  și  $j$  două valori naturale ( $1 \leq i \leq j \leq 100$ );
  - returnează suma  $A_i + A_{i+1} + \dots + A_j$ .
- (6p.)

- b) Fișierul text **NUMERE.IN** conține pe prima linie două numere naturale nenule  $n$  și  $m$  ( $1 \leq m \leq n \leq 100$ ), pe linia a doua  $n$  numere naturale din intervalul  $[1, 10000]$  și pe linia a treia alte  $m$  numere naturale din intervalul  $[1, 10000]$ . Pe fiecare linie numerele sunt separate prin câte un spațiu.

Scrieți un program `C/C++` care citește cele două numere naturale  $n$  și  $m$  din fișierul **NUMERE.IN**, construiește în memorie două tablouri unidimensionale **A** și **B** cu elementele aflate în fișier pe a doua, respectiv a treia linie și verifică, utilizând apeluri utile ale subprogramului **suma**, dacă tabloul **A** se poate reduce la tabloul **B**. Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz negativ. (4p.)