

Subiectul III (30 de puncte)

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

1. Un program citește o valoare naturală nenulă pentru n și apoi generează și afișează, în ordine descrescătoare lexicografic, toate combinațiile de n cifre care aparțin mulțimii $\{0,1\}$. Astfel, pentru $n=2$, combinațiile sunt afișate în următoarea ordine: 11, 10, 01, 00. Dacă se rulează acest program și se citește pentru n valoarea 8, imediat după combinația 10101000 va fi afișată combinația: (4p.)
- a. 01010111 b. 10100111 c. 10101001 d. 10100100

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

2. Funcția f are definiția alăturată. Scrieți 4 valori de apel pe care le poate avea n astfel încât, pentru cele 4 apeluri, corespunzătoare acestor valori, să se obțină 4 valori, distincte două câte două. (6p.)
- ```
int f(int n)
{
 if (n<=9) return 0;
 if (n%4==0) return 0;
 return 1+f(n-3);
}
```
3. Funcția `verif` primește prin intermediul a trei parametri, notați  $a$ ,  $b$  și  $c$ , trei valori naturale nenule, fiecare de maximum patru cifre. Funcția returnează valoarea 1 dacă cele trei valori pot constitui laturile unui triunghi și valoarea 0 în caz contrar.
- a) Scrieți definiția completă a funcției `verif`. (5p.)
- b) Scrieți un program C/C++ care citește de la tastatură șase valori naturale nenule, fiecare de maximum patru cifre, apoi verifică, utilizând apeluri utile ale funcției `verif`, dacă primele trei numere citite pot constitui laturile unui triunghi și dacă ultimele trei numere citite pot constitui laturile unui triunghi; în caz afirmativ, programul afișează pe ecran mesajul **congruente** dacă cele două triunghiuri sunt congruente sau mesajul **necongruente** dacă cele două triunghiuri nu sunt congruente; dacă cel puțin unul dintre cele două triplete de valori nu pot constitui laturile unui triunghi, programul va afișa pe ecran mesajul **nu**. (5p.)
4. Fișierul **BAC.DAT** conține pe prima linie, separate printr-un spațiu, două valori naturale  $n$  și  $m$  ( $2 \leq n \leq 1000$ ,  $2 \leq m \leq 1000$ ), pe a doua linie  $n$  valori întregi și pe a treia linie  $m$  valori întregi. Valorile de pe a doua și de pe a treia linie apar în fișier în ordine strict crescătoare, sunt separate prin câte un spațiu și au cel mult 4 cifre fiecare.
- Se cere afișarea pe ecran a două valori, dintre cele aflate în poziții consecutive pe a treia linie a fișierului, care determină intervalul închis în care se află un număr maxim de valori de pe a doua linie a fișierului. Se va utiliza o metodă eficientă din punct de vedere al timpului de executare și al spațiului de memorie utilizat. Se garantează că cel puțin un număr aflat pe a doua linie a fișierului aparține unuia dintre intervalele determinate de numerele de pe a treia linie a fișierului.
- Exemplu:** dacă fișierul **BAC.DAT** are conținutul
- ```
10 4
-1 1 3 4 5 6 10 15 16 117
0 1 9 20
```
- alăturat, programul va afișa: 1 9
- Explicație: cele patru numere de pe a treia linie a fișierului determină trei intervale: $[0,1]$, $[1,9]$, $[9,20]$; în intervalul $[1,9]$ se află 5 valori de pe a doua linie a fișierului, acesta fiind numărul maxim de valori aflate în unul dintre cele trei intervale.
- a) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)
- b) Scrieți un program C/C++ care să rezolve problema conform metodei descrise. (6p.)