

Subiectul III (30 de puncte)

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

1. Funcția F are definiția alăturată. Ce valoare are $F(18)$? (4p.)

```
int F(int x){  
    if (x<=1) return x;  
    else return x+F(x-2);  
}
```

- a. 90 b. 171 c. 91 d. 18

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

2. Un algoritm generează în ordine crescătoare toate numerele de n cifre ($n < 9$), cu cifre distincte, care nu au două cifre pare alăturate. Dacă pentru $n=5$, primele cinci soluții generate sunt 10325, 10327, 10329, 10345, 10347, precizați care sunt următoarele **trei** soluții generate, în ordinea obținerii lor. (6p.)

3. Subprogramul **aranjare** are doi parametri, a și n , prin care primește un tablou unidimensional cu maximum 100 de numere reale nenule și, respectiv, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului astfel încât toate valorile negative să se afle pe primele poziții, iar valorile pozitive în continuarea celor negative. Ordinea în cadrul secvenței de elemente pozitive, respectiv în cadrul secvenței de elemente negative, poate fi oricare.

Exemplu: dacă este transmis ca parametru un tablou unidimensional cu 6 elemente de forma (12, -7.5, 6.5, -3, -8, 7.5), după apel, acesta ar putea fi: (-7.5, -3, -8, 12, 6.5, 7.5).

a) Scrieți definiția completă a subprogramului **aranjare**. (10p.)

b) Scrieți un program C/C++ care citește de la tastatură un număr natural n ($1 \leq n \leq 100$) și apoi un șir de n numere reale nenule și care, folosind apeluri utile ale subprogramului **aranjare**, afișează pe ecran, separate prin spațiu, mai întâi elementele pozitive din șir și apoi cele negative. Ordinea în cadrul secvenței de elemente pozitive, respectiv în cadrul secvenței de elemente negative, poate fi oricare.

Exemplu: pentru $n=5$ și pentru șirul 6, -16.3, 8, -18, 20.7 se poate afișa pe ecran soluția
6 8 20.7 -18 -16.3 (4p.)

4. În fișierul **numere.txt** sunt memorate mai multe numere reale separate prin câte un spațiu. Scrieți un program C/C++ care verifică dacă printre numerele din fișier există cel puțin 10 numere naturale. Programul afișează pe ecran mesajul **DA** în caz afirmativ și **NU** în caz contrar.

Exemplu: dacă fișierul **numere.txt** conține numerele 60 -12.67 15 -1 -22.3 4 se afișează mesajul **NU**. (6p.)