

Subiectul III (30 de puncte)

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

1. Generarea tuturor șirurilor de caractere de 5 litere, fiecare literă putând fi orice element din mulțimea $\{p, r, o, b, a\}$, se realizează cu ajutorul unui algoritm echivalent cu algoritmul de generare a: (4p.)
- a. aranjamentelor b. permutărilor c. submulțimilor d. produsului cartezian

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

2. Se consideră subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului `f1(4);`? (6p.)
- ```
void f1(int x)
{
 if (x <= 9)
 {
 cout << x + 1; | printf("%d", x + 1) ;
 f1(x + 2);
 }
}
```
3. Scrieți în limbajul C/C++ definiția completă a subprogramului **ordonare** care primește ca parametru un tablou unidimensional **x** cu cel mult 100 de elemente, numere întregi cu cel mult 4 cifre fiecare, și un număr natural **n** ( $n \leq 100$ ), ce reprezintă numărul efectiv de elemente ale tabloului **x**. Subprogramul modifică tabloul prin schimbarea poziției doar a elementelor impare, astfel încât acestea să apară în tablou în ordinea crescătoare a valorilor lor. Subprogramul afișează pe ecran, separate prin câte un spațiu, elementele tabloului obținut, de la primul până la ultimul. Tabloul modificat va fi furnizat tot prin parametrul **x**. (10p.)

**Exemplu:** pentru  $n=6$  și  $x=(\underline{7}, \underline{11}, 2, -8, \underline{-3}, 10)$

subprogramul va afișa  $\underline{-3} \ \underline{7} \ 2 \ -8 \ \underline{11} \ 10$

4. În fișierul **numere.txt**, se află memorate, pe prima linie un număr natural **n** ( $1 \leq n \leq 100$ ), iar pe fiecare dintre următoarele **n** linii, câte două numere întregi **x, y** ( $-100 \leq x \leq y \leq 100$ ), reprezentând capetele câte unui segment **[x, y]** desenat pe axa **Ox** de coordonate.

a) Scrieți în limbajul C/C++ un program eficient din punct de vedere al timpului de executare și al spațiului de memorare, care citește din fișier datele existente, determină segmentul rezultat în urma intersecției tuturor celor **n** segmente date și afișează pe ecran două numere despărțite printr-un spațiu ce reprezintă capetele segmentului cerut. Dacă segmentele nu au niciun punct comun se va afișa pe ecran valoarea 0. (6p.)

b) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. (4p.)

**Exemplu:** dacă fișierul **numere.txt** are conținutul alăturat, se va afișa

|    |    |
|----|----|
| 5  |    |
| -7 | 10 |
| 3  | 20 |
| -5 | 5  |
| 0  | 12 |
| -8 | 30 |

pe ecran

3 5