

**Subiectul III (30 de puncte)**

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

1. Având la dispoziție cifrele 0, 1 și 2 putem genera, în ordine crescătoare, numere care au suma cifrelor egală cu 2 astfel încât primele 6 numere generate sunt, în această ordine: 2, 11, 20, 101, 110, 200. Folosind același algoritm se generează numere cu cifrele 0, 1, 2 și 3 care au suma cifrelor egală cu 4. Care va fi al 7-lea număr din această generare ? **(4p.)**
- a. 103                      b. 301                      c. 220                      d. 130

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

2. Se consideră subprogramul recursiv **f1** definit alăturat. Ce se va afișa în urma apelului **f1(5)**? **(6p.)**
- ```
procedure f1(x:integer);
begin
  if x<=9 then
  begin
    write(x+1); f1(x+2); write(x+3)
  end
end;
```
3. Scrieți definiția completă subprogramului **suma** care primește ca parametru un tablou unidimensional **x** cu cel mult 100 de elemente numere reale, un număr natural **n** ce reprezintă dimensiunea tabloului **x** ( $n \leq 100$ ), și un număr natural **m** ( $n \geq m$ ). Subprogramul returnează suma obținută din cele mai mici **m** elemente ale tabloului **x**. **(10p.)**
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 **Pascal** 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 nici un 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 pe ecran

3 5

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