

SUME – DESCRIEREA SOLUȚIEI

Problema pare deosebit de complexă. O primă idee, greu de implementat, este de a genera toate submulțimile de n elemente ale mulțimii $\{1, 2, \dots, n^2\}$, apoi de a alege n submulțimi disjuncte cu suma elementelor aceeași. Pe lângă faptul că această idee este greu de implementat, este și inutilizabilă din punct de vedere practic, numărul de submulțimi fiind foarte mare. Soluția se bazează pe o idee constructivă.

1. Plasăm în ordine elementele de la 1 la n^2 într-o matrice pătratică cu n linii și n coloane. De exemplu, pentru $n=4$ obținem:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

2. Alegem ca primă submulțime elementele de pe diagonala principală $\{1, 6, 11, 16\}$. Suma elementelor din submulțime este $S=1+6+11+16=34$.
3. Pentru a construi cu ușurință celelalte $n-1$ submulțimi cu suma elementelor S , vom extinde matricea, copiind în continuare primele $n-1$ linii. Se obține astfel o matrice cu $2*n-1$ linii și n coloane. De exemplu, pentru $n=4$ obținem:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
1	2	3	4
5	6	7	8
9	10	11	12

Această matrice are n pseudodiagonale principale. Mulțimile elementelor de pe oricare două pseudodiagonale sunt disjuncte, iar suma elementelor de pe fiecare pseudodiagonală este aceeași.

Problema se reduce astfel la o problemă asemănătoare cu problema precedentă.

```
#include <fstream.h>
#define InFile "sume.in"
#define OutFile "sume.out"
int main()
{
    ifstream fin (InFile);
    ofstream fout(OutFile);
    int n, a[201][101], nr=1, i, j, d;
    fin>>n;
    //construim tabloul
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            a[i][j]=a[i+n][j]=nr++;
    //parcurgem pseudodiagonalele
    for (d=0; d<n; d++)
    {
        //afisez elementele de pe pseudodiagonala d
        for (j=0; j<n-1; j++) fout<<a[j+d][j]<<' ';
        fout<<a[n-1+d][n-1]<<'\n';
    }
    fin.close(); fout.close();
    return 0;
}
```