# ***ALGORITMUL SUMMA***

# ***Varianta secvențială***

***Student : Lețu Andreea-Roxana***

***Anul : 3***

***Grupa : 2***

***Specializare : Calculatoare Română***

1. Cerință : Se va proiecta algoritmul SUMMA (vaiantă secvențială ) pentru înmulțirea a 2 matrici .

Pentru varianta secvențială am ales implementarea in Java .

Codul implementează înmulțirea matricelor folosind algoritmul **SUMMA** (Summation Algorithm), care împarte matricile mari în blocuri de matrici mai mici pentru a optimiza procesul de înmulțire.

**Pași principali:**

1. **Inițializarea matricelor**: Două matrici A și B sunt generate cu valori specifice.

Pentru fiecare element A[i][j], valoarea este calculată adunând **indicele rândului i, indicele coloanei j și 1.**

Pentru fiecare element B[i][j], valoarea este calculată ca produsul dintre **indicele rândului i plus 1** și **indicele coloanei j plus 1**.

1. **Împărțirea în blocuri**: Matricile sunt împărțite în blocuri de matrici mai mici pentru a îmbunătăți performanța prin reducerea accesului la memorie.
2. **Încheierea produsului**: Algoritmul SUMMA efectuează înmulțirea blocurilor și adună rezultatele în matricea C.
3. **Măsurarea timpului de execuție**: Timpul de execuție al algoritmului este măsurat pentru a evalua performanța.
4. **Afișarea rezultatelor**: Matricile A, B, și C sunt afișate, împreună cu timpul de execuție al algoritmului.

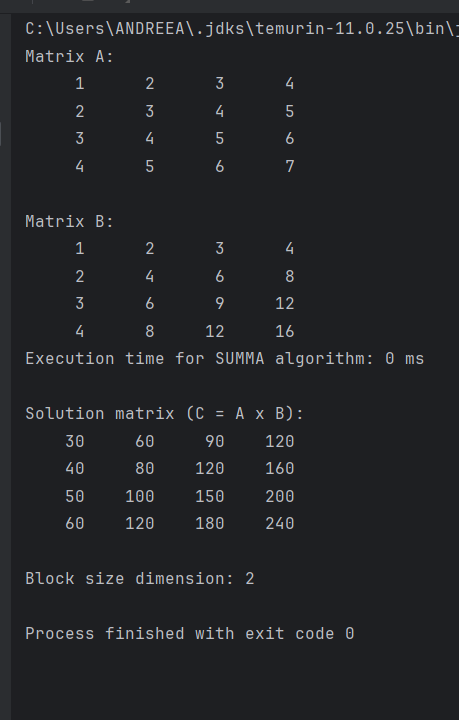
Codul este structurat pe mai multe fișiere pentru o mai bună organizare, fiecare responsabil de o parte a algoritmului: inițializare, înmulțire blocuri, măsurarea timpului .

2) Info despre masina pe care ati rulat codul

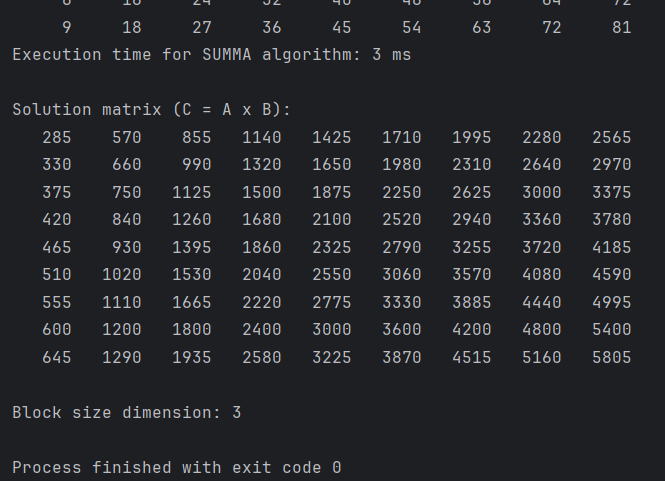
3)  Rezultatele experimentale, timpii de rulare.

Pentru următoarele rulări avem următoarele valori ale timpilor de execuție :

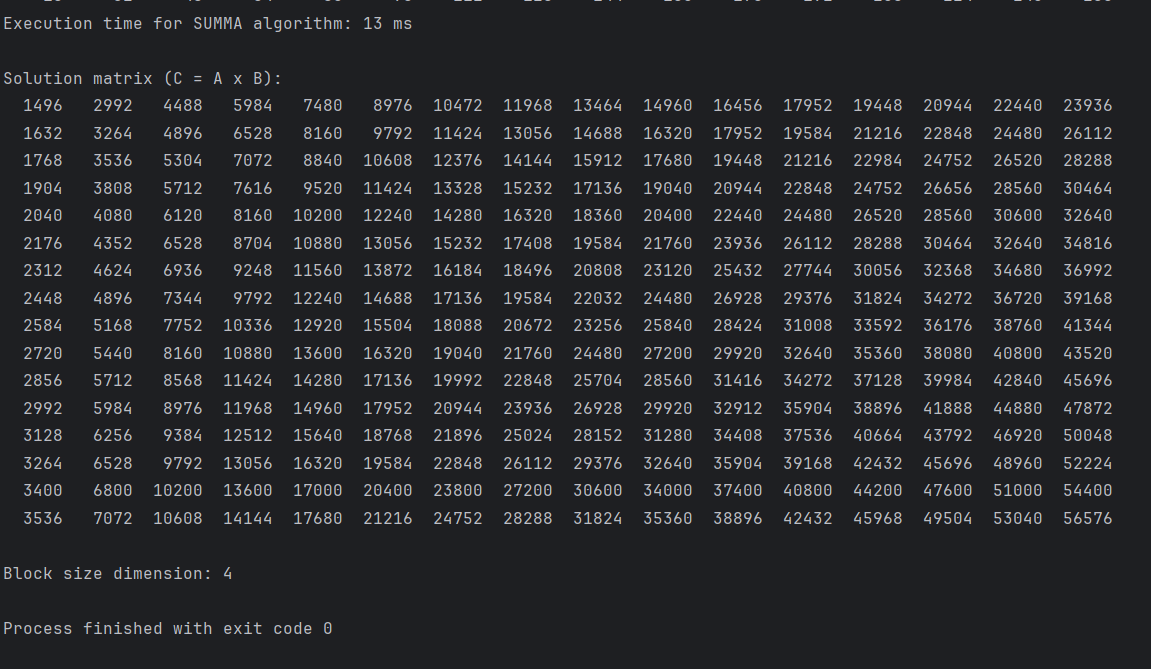
N=4



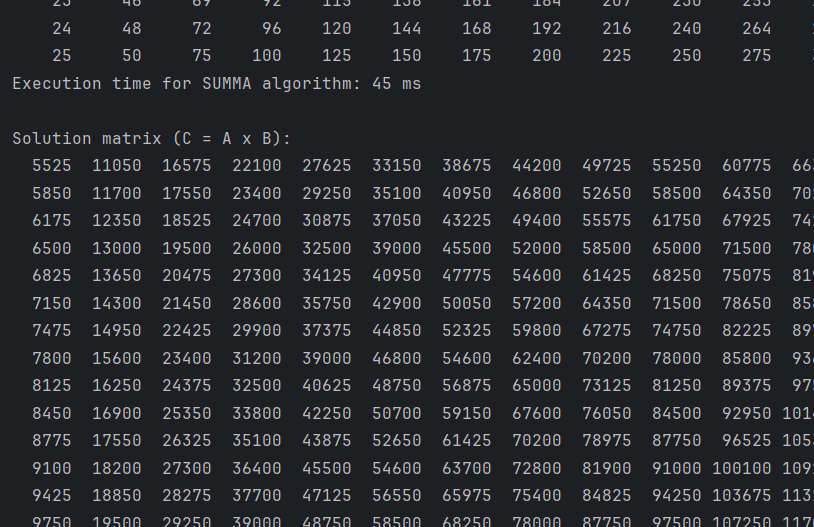
N=9



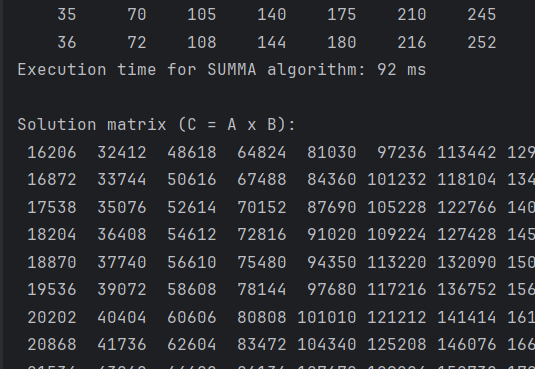
N=16



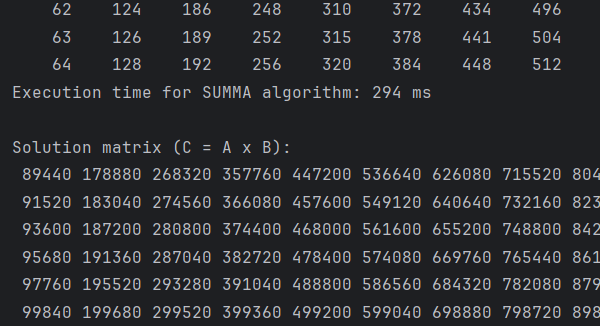
N=25



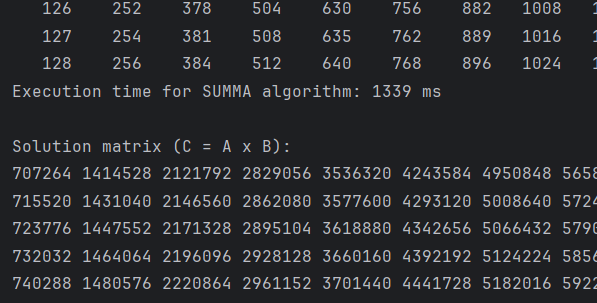
N= 36



N=64



N=128



Graficul de evolutie al executiei algoritmului Summa in functie de datele de intrare :

|  |  |
| --- | --- |
| Dimensiune - N | Timp de executie (ms) |
| 4 | 0 |
| 9 | 3 |
| 16 | 13 |
| 25 | 45 |
| 36 | 92 |
| 64 | 294 |
| 128 | 1339 |