1. Se dă matrice de tip char. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească numarul de vocale în coloana prelucrată. În main de afișat maximul din rezultatele firelor. De utilizat interfața Runnable.
2. Se dă matrice de tip short. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separat. Fiecare fir trebuie să găsească media aritmetică a tuturor numerelor negative în rîndul prelucrat. În main de afișat minimul din rezultatele firelor. De utilizat interfața Runnable.
3. Se dă matrice de tip boolean. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească numărul de elemente cu valori true în coloana prelucrată. În main de afișat suma a rezultatelor firelor. De utilizat clasa Thread.
4. Se dă matrice de tip byte. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separat. Fiecare fir trebuie să găsească media aritmetică a elementelor în rîndul prelucrat. În main de afișat suma a rezultatelor firelor. De utilizat clasa Thread.
5. Se dă matrice de tip boolean. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să calculeze ”&&” cu toate elementele din coloana prelucrată. În main de afișat calcularea ”&&” cu toate rezultatele din fiecare fir. De utilizat interfața Runnable.
6. Se dă matrice de tip String. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separate. Fiecare fir trebuie să găsească media aritmetică din lungimile elementelor în rîndul prelucrat. În main de afișat minimul din rezultatele firelor. De utilizat clasa Thread.
7. Se dă matrice de tip long. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească numărul de elemente negative în coloana prelucrată. În main de afișat produsul rezultatelor firelor. De utilizat interfața Runnable.
8. Se dă matrice de tip int. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separat. Fiecare fir trebuie să găsească media aritmetică a tuturor numerelor pozitive în rîndul prelucrat. În main de afișat media aritmetică a tuturor rezultatelor firelor. De utilizat clasa Thread.
9. Se dă matrice de tip char. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească numărul de operații aritmetice (‘+’, ‘-‘, ‘\*’, ‘/’) în coloana prelucrată. În main de afișat raportul dintre numărul total de operații aritmetice din matrice cu numărul total de elemente din matrice. De utilizat interfața Runnable.
10. Se dă matrice de tip float. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separate. Fiecare fir trebuie să găsească elementul maximum din toate elemente negative în rîndul prelucrat. În main de afișat elementul maxim din rezultatele firelor. De utilizat interfața Runnable.
11. Se dă matrice de tip double. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească media aritmetică a numerelor întregi în coloana prelucrată. În main de afișat elementul maxim din rezultatele firelor. De utilizat clasa Thread.
12. Se dă matrice de tip short. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească numărul elementelor impare în rîndul prelucrat. În main de afișat raportul dintre numărul total de elemente impare din matrice cu numărul total de elemente din matrice. De utilizat clasa Thread.
13. Se dă matrice de tip long. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească elementul minim din toate numerele pozitive în coloana prelucrată. În main de afișat media aritmetică a tuturor rezultatelor firelor. De utilizat interfața Runnable.
14. Se dă matrice de tip int. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separate. Fiecare fir trebuie să găsească numărul elementelor negative în rîndul prelucrat. În main de afișat suma a rezultatelor firelor. De utilizat clasa Thread.
15. Se dă matrice de tip byte. De divizat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separate. Fiecare fir trebuie să găsească produsul tuturor elementelor nenegative în rîndul prelucrat. În main de afișat elementul minim din rezultatele firelor. De utilizat interfața Runnable.
16. Se dă matrice de tip double. De divizat matricea pe coloane, fiecare coloană fiind atribuită unui fir de execuție separat. Fiecare fir trebuie să găsească în coloana prelucrată numărul elementelor mai mari decît x, unde x – număr introdus de la tastatură. În main de afișat elementul maxim din rezultatele firelor. De utilizat clasa Thread.
17. Se dă matrice de tip int. De separat matricea pe rînduri. fiecare rînd fiind atribuit unui fir de execuție separate. Fiecare fir trebuie să găsească în rind propriu media aritmetică a tuturor elemelntelor mai mici ca x, unde x - numar introdus de la tastatură. În main de afișat elementul maxim din rezultatele firelor. De utilizat clasa Thread.
18. De scris un program în Java care, pentru o matrice bidimensională, în locul elementelor scrie suma vecinilor acestora, numărul de iterații și matricea se introduce de la tastatură. În program de folosit cîte un fir de execuție, care va face calcule, pentu fiecare linie a matricei. De afișat punctul de inceput și sfirșit pentru fiecare fir, precum și momentul de rescriere a matricei. Clasa firelor de execuție trebuie să fie moștenită de la Thread.