

Задачи за прости действия с матрици - равенство, събиране, изваждане, умножение с число, транспониране

Всичките ни матрици ще са с размер по второто направление 50 ( $n \times 50$ )

Обяснението защо е свързано с това как се пазят многомерните масиви в паметта

1. Напишете `bool mEqual(int A[][50], int n1, int m1, int B[][50], int n2, int m2)`, където `A` и `B` са статични масиви, и която връща дали матриците са равни
2. Напишете функция `sum(int A[][50], int B[][50], int C[][50], int n, int m)`, която приема две “матрици” - `A` и `B` и записва в `C` сумата им
3. Напишете функция `transpose(int A[][50], int B[][50], int n, int m)`, която записва в `B` транспонираната матрица на `A` (ако `A` е от тип  $n \times m$ , то `B` ще бъде от  $m \times n$ )

(Примерът долу предполага, че в `main` има въвеждане на размери и клетки за матрицата `A` (до  $50 \times 50$ ), последвано от извикване на транспониращата функция, последвано от изкарването на резултата на екрана)

Примерен вход	Примерен изход
3 3 9 32 -209 78 -12 0 42 18 -1234	9 78 42 32 -12 18 -209 0 -1234

4. Напишете функция `sMult(int A[][50], int R[][50], int n, int m, int s)`, която да записва в `B` резултата от умножението на `A` с `s`
5. Използвайки `sum` и `sMult`, напишете функция `sub(int A[][50], int B[][50], int C[][50], int n, int m)`, която записва в `C` резултатът от `A-B`
6. Напишете функция `matrixMult(int A[][N], int B[][N], int C[][N], int r1, int c1, int r2, int c2)`, която намира произведението на матриците `A` и `B` и записва резултата в `C`.