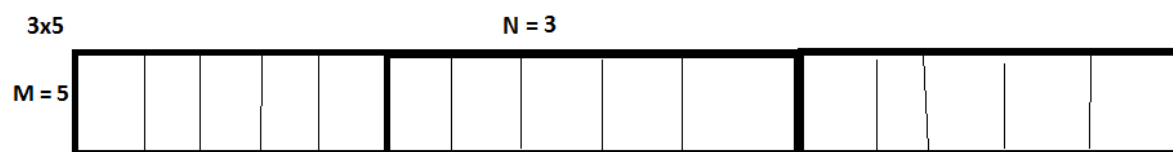


Проблем:

int A[3][5]



Моделът на паметта в C++ за масиви е непрекъснат в паметта блок от елементи, всеки от който е на $i \cdot \text{element_size}$ (1) отстъп от началото на масива.

Проблемът, който представят пред нас многомерните масиви е, че всеки техни елемент е друг масив, с фиксирана памет - размерът на тази памет е $j \cdot \text{element_size} \cdot \text{elements_in_array}$ (2). `elements_in_array` обаче е неизвестно и трябва да бъде указано някъде - затова се налага да подадем размера на масива по едното направление, когато подаваме двумерен масив

- (1) където i е индексът на елемента в масива, а `element_size` е размерът на един произволен елемент
- (2) Където j е индексът на едномерен масив, който "живее" в нашия двумерен масив, `element_size` е размерът на един елемент, `elements_in_array` е броят на елементите, които има в един от едномерните масиви, които "живеят" в нашият двумерен масив

C99 (https://en.wikipedia.org/wiki/Variable-length_array):

Стандартът на C позволява заделянето на допълнителна памет на стека за масиви с динамична дължина. Или иначе казано, можем да правим "статични" масиви с размер големината на променлива

C++:

Можем да направим константа `N`, на която да даваме фиксирана стойност и да я използваме за единия(втория) размер на всички двумерни масиви, които за които викаме функциите си - това налага ограничение отгоре на размера по едното измерение на всички матрици, които ползваме - то ни трябва единствено, за да може да работи аритметиката на указателите вътре - можем да използваме по-малко колони, ако искаме