

# Информатика

---

Кочанов Марк

1 декабря 2018 г.

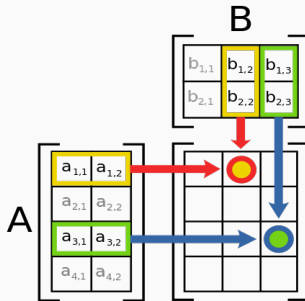
МИФИ

# Умножение матриц

Дана матрица  $A$  размера  $n \times m$  ( $n$  строк,  $m$  столбцов), задаваемая набором чисел  $a_{ij}$  (первый индекс — номер строки, второй — столбца).

Даны две матрицы:  $A$  ( $n \times m$ ) и  $B$  ( $m \times l$ ). Тогда определена операция умножения двух матриц (порядок умножения матриц фиксирован), дающая третью матрицу  $C = AB$  ( $n \times l$ ). Элементы матрицы  $C$  имеют вид (строка первой матрицы умножается «скалярно» на столбец второй матрицы):

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}.$$



# Задача

---

# Матрицы

Реализовать программу для работы с матрицами целых чисел. Программа должна использовать структуры для хранения матрицы. Для хранения элементов матрицы использовать двумерный динамический массив (массив массивов), либо одномерный массив (упаковка матрицы в вектор по столбцам или строкам).

Программа должна включать следующие функции: функция создания матрицы заданной размерности (без инициализации); функция создания и инициализации матрицы нулями; функция создания матрицы случайных чисел в заданном диапазоне; функция очистки памяти, занимаемой матрицей; функция транспонирования матрицы; функция сложения двух матриц, функция перемножения двух матриц; функция умножения матрицы на скаляр (целое число). При необходимости функция должна возвращать статус (например, отрицательная размерность, перемножение матриц неверных размерностей).

# Прототипы

```
1  typedef int dim;
2  typedef struct matrix {dim row, dim columns, ... } matrix;
3  typedef enum state { OK, WRONG_DIM, NEG_DIM, ... } state;
4
5  state m_init(dim rows, dim cols);
6  state m_create_zero(dim rows, dim cols);
7  state m_create_random(dim rows, dim cols, int min, int max);
8  void m_free(matrix* m);
9  matrix m_transpose(matrix* m);
10 state m_m_sum(matrix* m1, matrix* m2, matrix *m_res);
11 state m_m_multiply(matrix* m1, matrix* m2, matrix *m_res);
12 state m_s_multiply(matrix* m, int s, matrix *m_res);
```

В функции `m_create_zero` использовать функцию `memset`, определенную в заголовочной файле `string.h`

```
1  // ptr указатель на начало памяти
2  // value значение для инициализации
3  // num количество байт, которые необходимо проинициализировать
4  void* memset(void* ptr, int value, size_t num);
```