

Лекция 11

Программирование матриц

Преподаватель Палехова Ольга Александровна,
кафедра О7 БГТУ «Военмех»

Матрица

Матрица – математический объект, записываемый в виде прямоугольной таблицы элементов.

Матрица представляет собой совокупность **строк** и **столбцов**, на пересечении которых находятся её элементы. Элемент, находящийся на пересечении строки i со столбцом j обозначается a_{ij} .

Количество строк и столбцов задает **размер** матрицы, например, матрица $n \times m$.

Программирование матриц

Для программирования матриц на языке Си используют:

- **одномерные массивы:**
 - статические,
 - динамические;
- **двумерные массивы:**
 - статические,
 - динамические;
- **динамические матрицы.**



Матрица может храниться по строкам или по столбцам. Выбор способа хранения зависит от используемых алгоритмов.

Размер и размерность массива

Размер массива – количество элементов в нем ($A(N)$, $B(M \times N)$, $Q(M \times N \times K)$).

Размерность массива – количество индексов, необходимых для обращения к элементу:

- одномерные массивы (векторы) – **$a[i]$**
- двумерные массивы – **$a[i][j]$**
- трехмерные массивы – **$a[i][j][k]$**
- четырехмерные массивы – **$a[i][j][k][q]$**



В языке Си массивы любой размерности моделируются с помощью одномерных массивов.

Ввод и вывод прямоугольной матрицы

Ввод с клавиатуры:

```
for ( i = 0; i < M; i++ )  
    for ( j = 0; j < N; j++ )  
        scanf ( "%d", &a[i][j] );
```

цикл по строкам

цикл по столбцам

Вывод на экран:

```
for ( i = 0; i < M; i++ )  
{  
    for ( j = 0; j < N; j++ )  
        printf ( "%5d", a[i][j] );  
    printf ( "\n" );  
}
```

цикл по строкам

вывод строки

в той же строке

перейти на
новую строку



Ввод и вывод матриц обычно осуществляют по строкам.

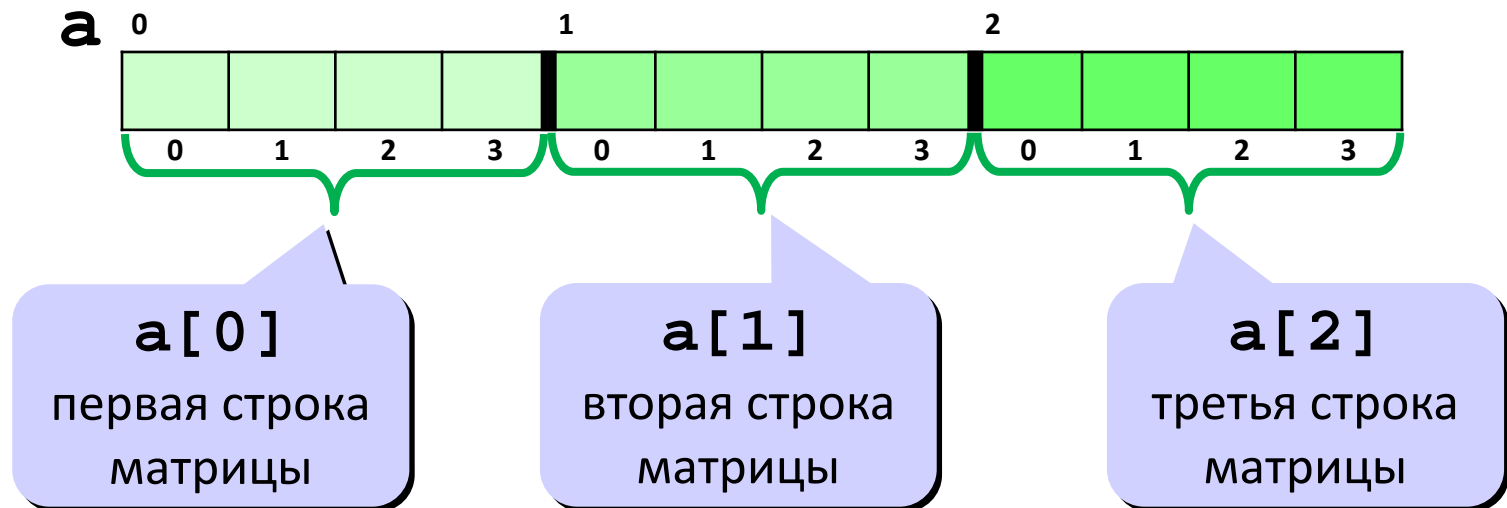
Двумерный массив в Си

В Си **двумерный массив** моделируется с помощью массива, элементами которого являются массивы.

Объявление статического двумерного массива

```
int a[3][4];
```

используется для программирования матрицы 3x4, но является массивом из 3 элементов, элементы которого – массивы из 4 целых чисел.



Статический двумерный массив

Особенности:

- память выделяется **одним блоком**;
- двумерный массив – это **массив массивов**
- имя массива – **указатель-константа на первый элемент** (первый подмассив);
- для обращения к элементу нужны **два индекса**;
- **может инициализироваться** полностью или частично;
- как правило, используется для хранения матрицы **по строкам**
- выход за пределы «строки» является **переходом к следующей «строке»**

Инициализация двумерных массивов

```
int a[2][2] = {{1,2},{5,0}};  
int b[2][3] = {{1,2},{-1}};  
double q[4][5] = {{1.1,1.2},  
                  {},  
                  {3.1,3.2,3.3}};
```

a

1	2
5	0

b

1	2	0
-1	0	0

q

1.1	1.2	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
3.1	3.2	3.3	0.0	0.0
0.0	0.0	0.0	0.0	0.0

x

1	2	3
2	7	5

y

1	2	3
2	7	5

```
int x[][3] = {{1,2,3},{2,7,5}};  
int y[][3] = {1,2,3,2,7,5};
```



Пустыми можно оставить только первые [] !

Статический двумерный массив

Задача: Дана матрица размером 4x4. Дополнить матрицу строкой, элементы которой равны суммам значений элементов соответствующих столбцов.

```
#include <stdio.h>
#define M 5
#define N 4
int main()
{
    int a[M][N], j, i;
    for ( i = 0; i < M-1 ; i++ )
        for ( j = 0; j < N; j++ )
            scanf ("%d", &a[i][j]);
    for ( j = 0; j < N; j++ )
    {
        a[M-1][j] = 0;
        for ( i = 0; i < M-1; i++ )
            a[M-1][j] += a[i][j];
    }
    /* вывод матрицы MxN */
    return 0;
}
```

число строк и столбцов
фиксировано

заполнение
по строкам

обработка
по столбцам

Статический двумерный массив

Задача: Дана матрица размером $M \times N$. Поменять местами строки с индексами k и v .

```
for ( j = 0; j < N; j++ )  
{  
    tmp = a[k][j];  
    a[k][j] = a[v][j];  
    a[v][j] = tmp;  
}
```

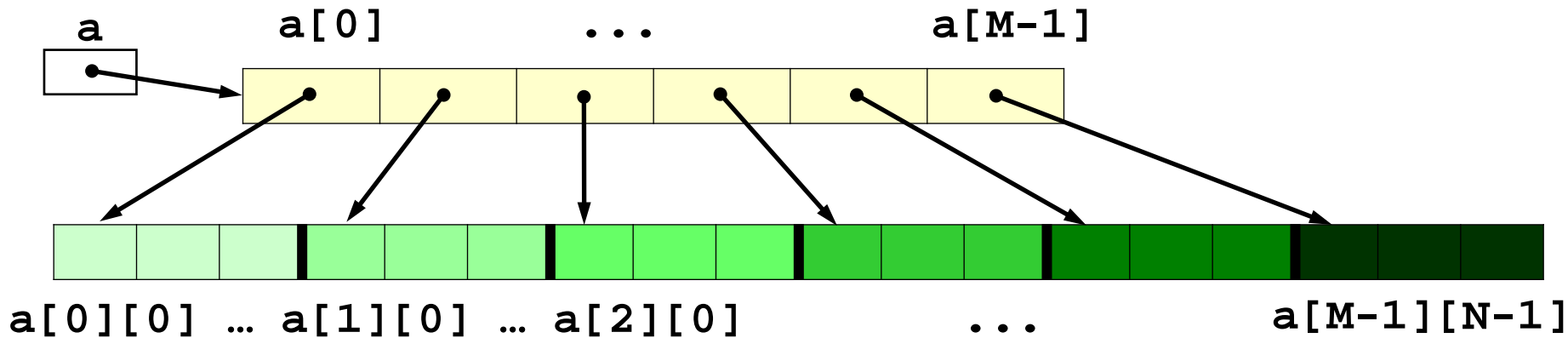
меняем строки
поэлементно

Задача: Дана матрица размером $M \times N$. Удалить из матрицы k -ю строку.

```
for ( i = k - 1; i < M - 1; i++ )  
    for ( j = 0; j < N; j++ )  
        a[i][j] = a[i+1][j];
```

сдвигаем
строки
поэлементно

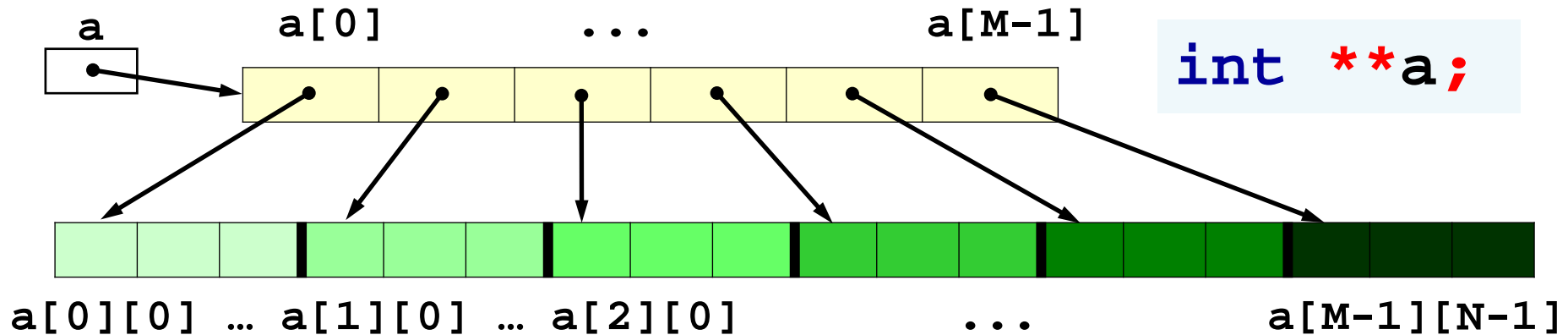
Динамический двумерный массив



Особенности:

- выделение памяти происходит **в процессе работы** программы
- память для данных выделяется **одним блоком**
- дополнительно требуется **массив указателей**
- имя массива – **указатель на указатель**;
- для обращения к элементу нужны **два индекса**;
- как правило, используется для хранения матрицы **по строкам**
- выход за пределы «строки» является **переходом к следующей «строке»**

Динамический двумерный массив



Выделение памяти:

```
a = calloc ( M, sizeof(int*));  
a[0] = calloc ( M*N, sizeof(int));
```

Расстановка указателей:

```
for ( i = 1; i < M; i++ )  
    a[i] = a[i-1] + N;
```

Освобождение памяти:

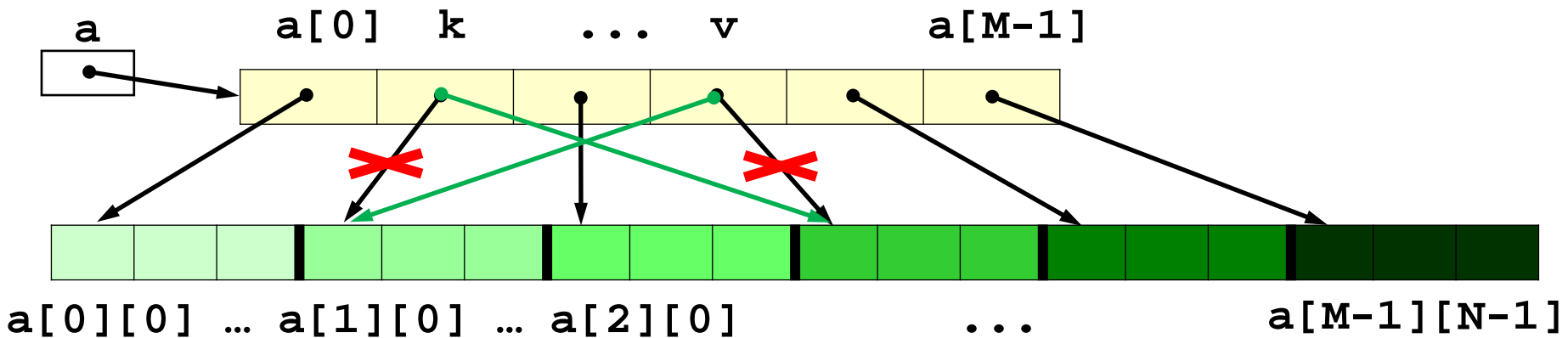
```
free (a[0]);  
free (a);
```



Как объявляется переменная `a`?

Динамический двумерный массив

Задача: Дана матрица размером $M \times N$. Поменять местами строки с индексами k и v .



```
tmp = a[k];  
a[k] = a[v];  
a[v] = tmp;
```

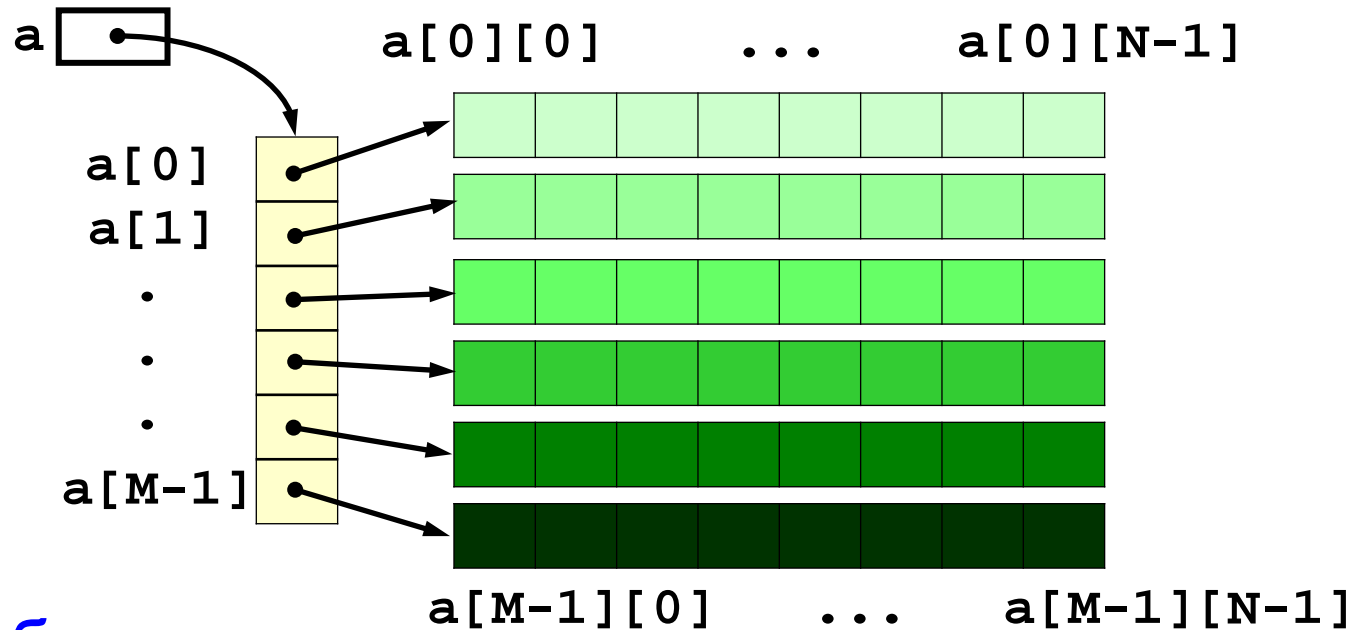
достаточно
переставить указатели



Значение `a[0]` изменять нельзя!

Удаление строки из матрицы выполняем в цикле так же, как и при использовании статической матрицы.

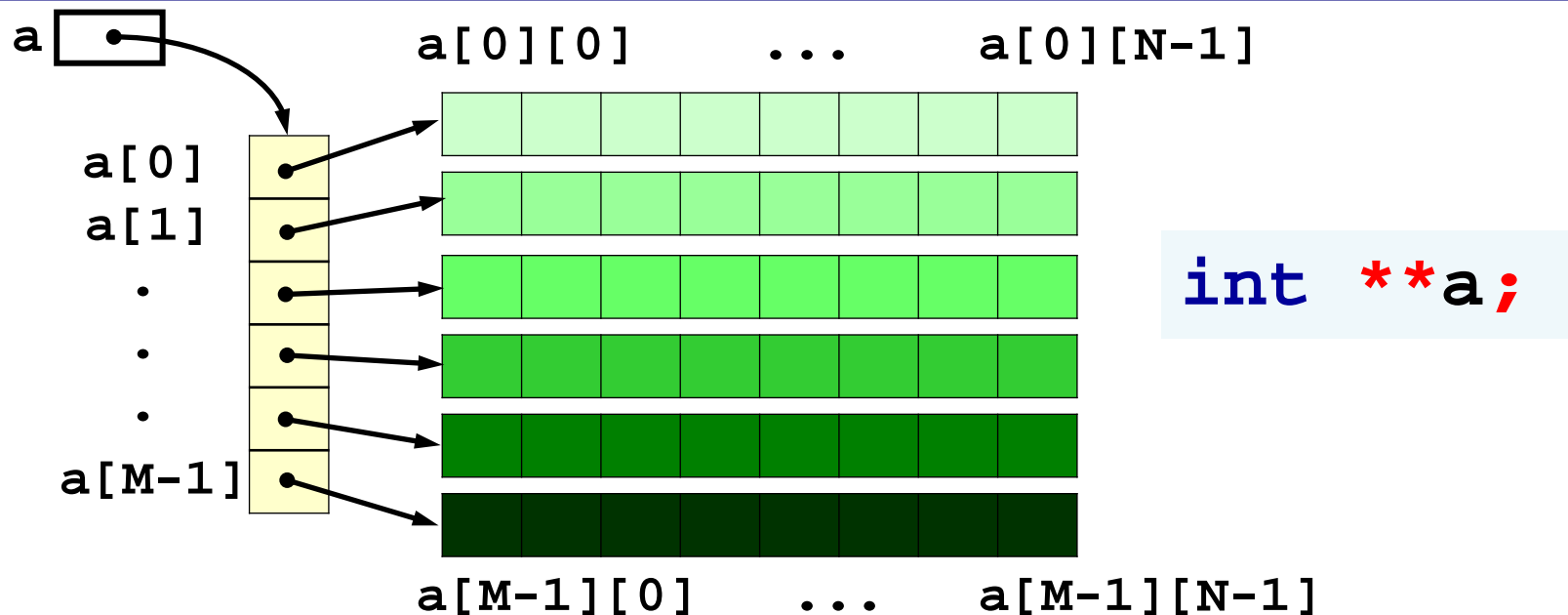
Динамическая матрица



Особенности:

- выделение памяти происходит **в процессе работы** программы
- память для данных выделяется **несколькими блоками**
- дополнительно требуется **массив указателей**
- имя массива – **указатель на указатель**;
- для обращения к элементу нужны **два индекса**;
- выход за пределы «строки» - **ОШИБКА**

Динамическая матрица



Выделение памяти:

```
a = malloc ( M*sizeof(int*));  
for ( i = 0; i < M; i++ )  
    a[i] = calloc ( N, sizeof(int));
```

Освобождение памяти:

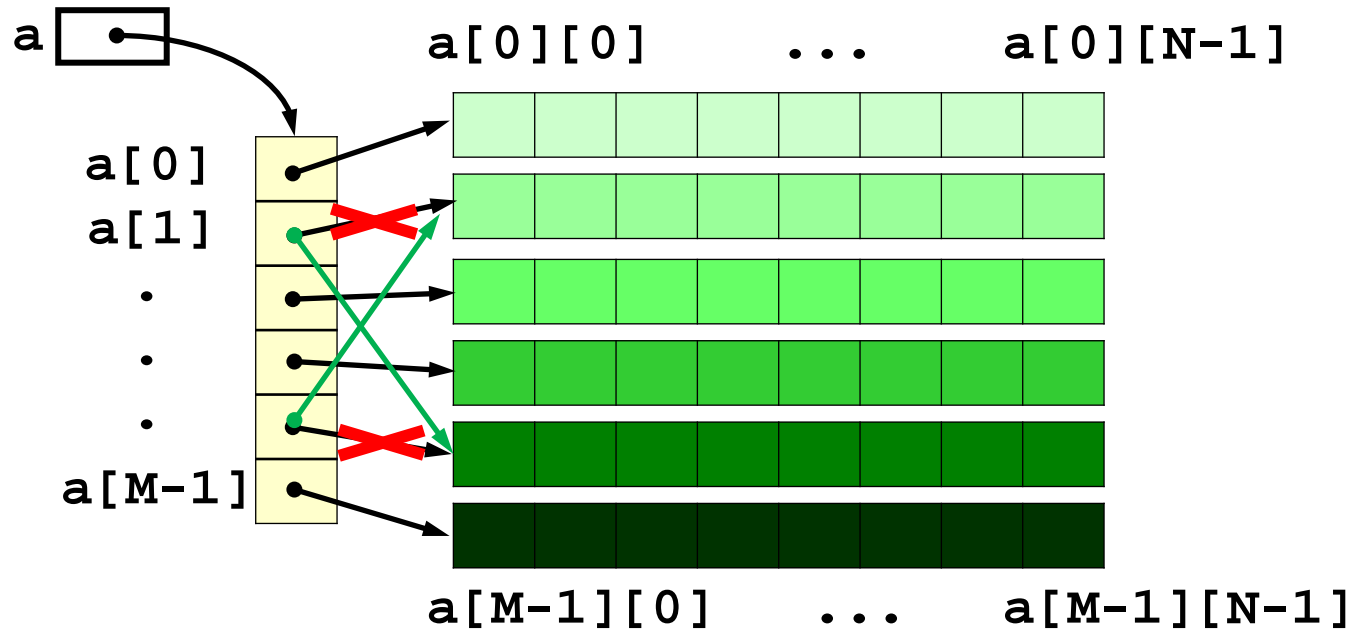
```
for ( i = 0; i < M; i++ )  
    free (a[i]);  
free (a);
```



Как объявляется
переменная `a`?

Динамическая матрица

Задача: Дана матрица размером $M \times N$. Поменять местами строки с индексами k и v .

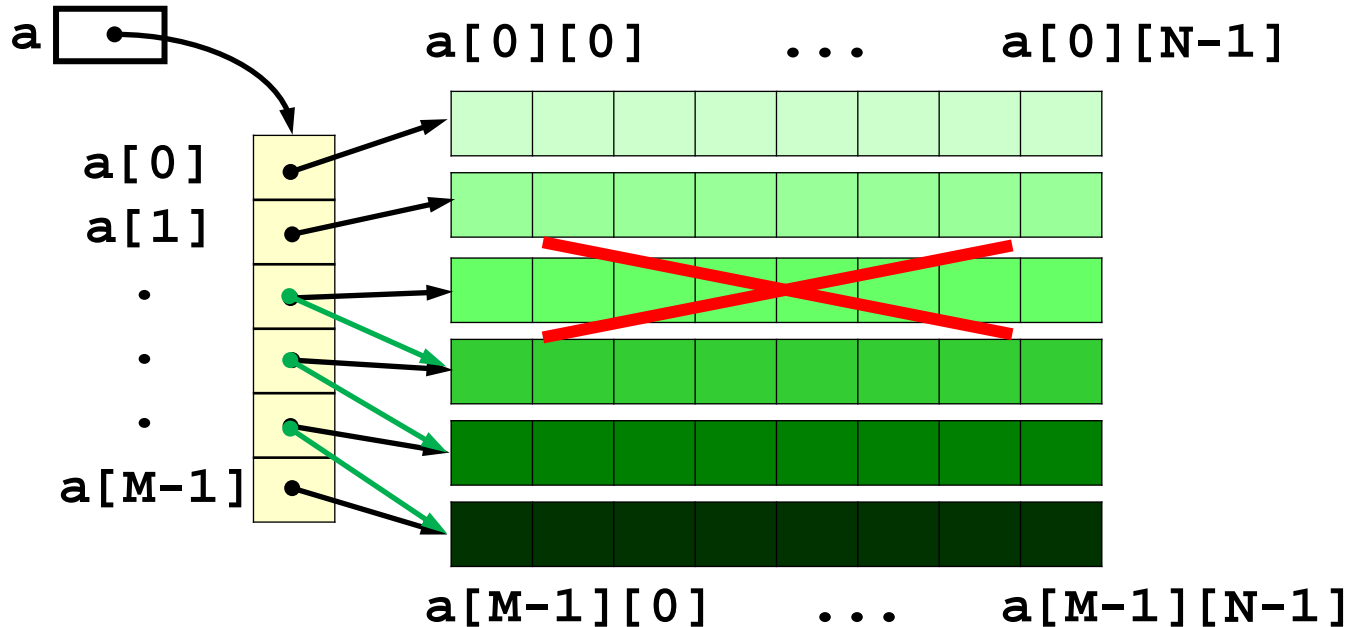


```
tmp = a[k];  
a[k] = a[v];  
a[v] = tmp;
```

достаточно
переставить указатели

Динамическая матрица

Задача: Дана матрица размером $M \times N$. Удалить из матрицы k -ю строку.



```
free (a[k-1]);  
for ( i = k - 1; i < M - 1; i++ )  
    a[i] = a[i+1];  
a[M-1] = NULL;
```

удаляем строку

переставляем
указатели