

// 1. ОБЪЯВЛЕНИЕ И ИНИЦИАЛИЗАЦИЯ

```
int matrix[3][4];           // 3 строки, 4 столбца
float table[2][3];          // 2 строки, 3 столбца
```

// Явная инициализация

```
int arr[2][3] = {
    {1, 2, 3},    // Строка 0
    {4, 5, 6}     // Строка 1
};
```

// Неполная инициализация (остальные элементы = 0)

```
int arr2[3][3] = {
    {1, 2},
    {3, 4, 5},
    {6}        // Последний элемент строки = 0
};
```

// 2. ОБРАЩЕНИЕ К ЭЛЕМЕНТАМ

```
matrix[0][0] = 10;    // Левый верхний угол
x = matrix[1][2];     // Вторая строка, третий столбец
matrix[i][j] = value; // Запись в строку i, столбец j
```

// 3. ОБХОД МАССИВА (ВЛОЖЕННЫЕ ЦИКЛЫ)

// По строкам

```
for(int i = 0; i < rows; i++) {    // По строкам
    for(int j = 0; j < cols; j++) { // По столбцам
        printf("%d ", matrix[i][j]);
    }
    printf("\n"); // Новая строка после каждой строки массива
}
```

// По столбцам

```
for(int j = 0; j < cols; j++) {    // По столбцам
    for(int i = 0; i < rows; i++) { // По строкам
        printf("%d ", matrix[i][j]);
    }
    printf("\n");
}
```

// 4. ВВОД И ВЫВОД ДВУМЕРНОГО МАССИВА

// Ввод с клавиатуры

```
for(int i = 0; i < rows; i++) {
    for(int j = 0; j < cols; j++) {
        printf("matrix[%d][%d] = ", i, j);
        scanf("%d", &matrix[i][j]);
    }
}
```

// Вывод в виде таблицы

```
for(int i = 0; i < rows; i++) {
```

```

        for(int j = 0; j < cols; j++) {
            printf("%4d", matrix[i][j]); // Выравнивание
        }
        printf("\n");
    }

```

// 5. БАЗОВЫЕ АЛГОРИТМЫ

// Сумма всех элементов

```

int sum = 0;
for(int i = 0; i < rows; i++) {
    for(int j = 0; j < cols; j++) {
        sum += matrix[i][j];
    }
}

```

// Поиск максимума

```

int max = matrix[0][0];
for(int i = 0; i < rows; i++) {
    for(int j = 0; j < cols; j++) {
        if(matrix[i][j] > max) {
            max = matrix[i][j];
        }
    }
}

```

// Сумма элементов строки

```

for(int i = 0; i < rows; i++) {
    int row_sum = 0;
    for(int j = 0; j < cols; j++) {
        row_sum += matrix[i][j];
    }
    printf("Сумма строки %d: %d\n", i, row_sum);
}

```

// Сумма элементов столбца

```

for(int j = 0; j < cols; j++) {
    int col_sum = 0;
    for(int i = 0; i < rows; i++) {
        col_sum += matrix[i][j];
    }
    printf("Сумма столбца %d: %d\n", j, col_sum);
}

```

// 6. РАБОТА С ГЛАВНОЙ ДИАГОНАЛЬЮ (для квадратных матриц)

// Элементы главной диагонали: matrix[0][0], matrix[1][1], ...

```

for(int i = 0; i < n; i++) {
    printf("%d ", matrix[i][i]); // Главная диагональ
}

```

// 7. ЗАПОЛНЕНИЕ СЛУЧАЙНЫМИ ЧИСЛАМИ

```
#include <stdlib.h>
#include <time.h>

srand(time(NULL));
for(int i = 0; i < rows; i++) {
    for(int j = 0; j < cols; j++) {
        matrix[i][j] = rand() % 100; // Числа от 0 до 99
    }
}
```

// 8. ПОЛЕЗНЫЕ КОНСТАНТЫ

```
#define ROWS 5
```

```
#define COLS 4
```

```
int table[ROWS][COLS]; // Использование констант
```