

# 8. Multi-dimensional Arrays

## 8.1 Two-Dimensional Arrays

### 8.1.1 Declaration and Initialization

```
// Declaration
int matrix[3][4]; // 3 rows, 4 columns

// Initialization methods
int grid[2][3] = {
    {1, 2, 3},
    {4, 5, 6}
};

// Alternative initialization
int numbers[2][3] = {{1, 2, 3}, {4, 5, 6}};

// Partial initialization
int scores[3][2] = {
    {95, 87},
    {92, 78},
    {88} // Last element becomes 0
};
```

### 8.1.2 Accessing 2D Array Elements

```
int matrix[3][4] = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
};

// Accessing elements
printf("Element at row 1, column 2: %d\n", matrix[1][2]); // Output: 7

// Modifying elements
```

```
matrix[0][0] = 100;  
matrix[2][3] = 999;
```

### 8.1.3 Input/Output for 2D Arrays

```
#include <stdio.h>  
  
void input_matrix(int matrix[][4], int rows) {  
    printf("Enter matrix elements:\n");  
    for (int i = 0; i < rows; i++) {  
        for (int j = 0; j < 4; j++) {  
            printf("Element [%d][%d]: ", i, j);  
            scanf("%d", &matrix[i][j]);  
        }  
    }  
}  
  
void print_matrix (int matrix[][4], int rows) {  
    printf("Matrix:\n");  
    for (int i = 0; i < rows; i++) {  
        for (int j = 0; j < 4; j++) {  
            printf("%d ", matrix[i][j]);  
        }  
        printf("\n");  
    }  
}
```

---

Revision #1

Created 2025-09-15 01:03:48 UTC by DS

Updated 2025-09-15 01:04:13 UTC by DS