Programming Fundamentals

Muhammad Ateeq

[Updated: 01-04-2023]

2D Arrays

What are 2-dimensional arrays and how they relate to matrices?

In C++, a 2D array is an array of arrays. It is a way to represent a matrix, which is a 2D array of numbers. To declare a 2D array in C++, you can use the following syntax:

```
int matrix[row_size][column_size];
```

where row_size and column_size are the dimensions of the matrix. For example, to declare a 3x3 matrix, you can write:

```
int matrix[3][3];
```

You can access the elements of the matrix using the indices of the row and the column. For example, to access the element in the second row and third column, you can write:

```
matrix[1][2];
```

The first index (1) represents the row, and the second index (2) represents the column. You can use loops to traverse the elements of a matrix. For example, to print the elements of a 3x3 matrix, you can write:

```
for(int i=0; i<3; i++) {
    for(int j=0; j<3; j++) {
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}
```

This will print the matrix in row-major order, with each row on a separate line. Overall, a 2D array in C++ is a convenient way to represent matrices, and you can perform matrix operations using loops and other programming constructs.

Examples

Here are a few examples of 2D arrays in C++:

Example 1: Creating and Initializing a 2D Array

```
#include <iostream>
using namespace std;

int main() {
    // Create a 2D array of size 3x3
    int arr[3][3] = { {1, 2, 3},
                      {4, 5, 6},
                      {7, 8, 9} };

    // Print the elements of the array
```

```
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

Output:

```
1 2 3
4 5 6
7 8 9
```

Example 2: Taking Input from User for a 2D Array

```
#include <iostream>
using namespace std;

int main() {
    // Create a 2D array of size 3x3
    int arr[3][3];

    // Take input from the user for the array elements
    cout << "Enter the elements of the array: " << endl;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cin >> arr[i][j];
        }
    }

    // Print the elements of the array
    cout << "The elements of the array are: " << endl;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

Sample Output:

```
Enter the elements of the array:
1 2 3
4 5 6
7 8 9
The elements of the array are:
1 2 3
```

```
4 5 6
7 8 9
```

Example 3: Accessing Specific Element of a 2D Array

```
// Access and print the element in the second row and third column
cout << "The element at row 2, column 3 is: " << arr[1][2] << endl;
```

Example 4: Copying Elements of one 2D Array to Another

```
#include <iostream>
using namespace std;

int main() {
    // Create two 2D arrays of size 3x3
    int arr1[3][3] = { {1, 2, 3},
                       {4, 5, 6},
                       {7, 8, 9} };
    int arr2[3][3];

    // Copy the elements of arr1 to arr2
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            arr2[i][j] = arr1[i][j];
        }
    }

    // Print the elements of arr2
    cout << "The elements of arr2 are: " << endl;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr2[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

Output:

```
The elements of arr2 are:
1 2 3
4 5 6
7 8 9
```

Example 5: Finding Maximum Element in a 2D Array

```
#include <iostream>
using namespace std;

int main() {
```

```
    // Create a 2D array of size 3x3
    int arr[3][3] = { {10, 20, 30},
                      {40, 50, 60},
                      {70, 80, 90} };

    // Find the maximum element in the array
    int max = arr[0][0];
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (arr[i][j] > max) {
                max = arr[i][j];
            }
        }
    }

    // Print the maximum element
    cout << "The maximum element in the array is: " << max << endl;

    return 0;
}
```

Output:

```
The maximum element in the array is: 90
```

These additional examples should help you understand how to work with 2D arrays in C++ in more depth.