Data Structure Lab

Lab # 02

Double Pointer And

2D Array

Instructor:

Muhamma Saad Khan

Email:

saad.khan@nu.edu.pk

Course Code: CL2001

Department of Computer Science,

National University of Computer and Emerging Sciences FAST

Peshawar Campus

Double Pointer:

Now, we already know that a pointer stores the memory address of other variables. So, when we define a pointer to a pointer, the first pointer is used to store the address of the variables, and the second pointer stores the address of the first pointer. For this very reason, this is known as a Double Pointer or Pointer to Pointer.

The below diagram explains the concept of Double Pointers:

In a double pointer, Pointer 1 stores the address of a variable, and Pointer 2 stores the address of Pointer 1. This setup is called a pointer to pointer or double pointer.

How to Declare a Pointer to a Pointer in C ++?

Declaring a Pointer to Pointer is similar to declaring a pointer in C++. The difference is we have to use an additional * operator before the name of a Pointer in C++.

Syntax of a Pointer to Pointer(Double Pointer) in C++:

data_type_of_pointer **name_of_variable = & normal_pointer_variable;

```
// C++ program to implement
// pointer to pointer
#include <iostream >
using namespace std;
```

```
// Driver code
int main()
{
  int variable = 169;

  // Pointer to store the address
  // of variable
  int* pointer1;

  // double pointer to store the
  // address of pointer1
  int** pointer2;

  // Storing address of variable
  // in pointer1
  pointer1 = &variable;

  // Storing address of pointer1
  // in pointer2
  pointer2 = &pointer1;

  // Displaying the value of variable
  // with using both single and double
  // pointers.
  cout << "Value of variable :- " <<
          variable << "\n";
  cout << "Value of variable using single pointer :- " <<
          *pointer1 << "\n";
  cout << "Value of variable using double pointer :- " <<
          **pointer2 << "\n";
  return 0;
}
```

Output:

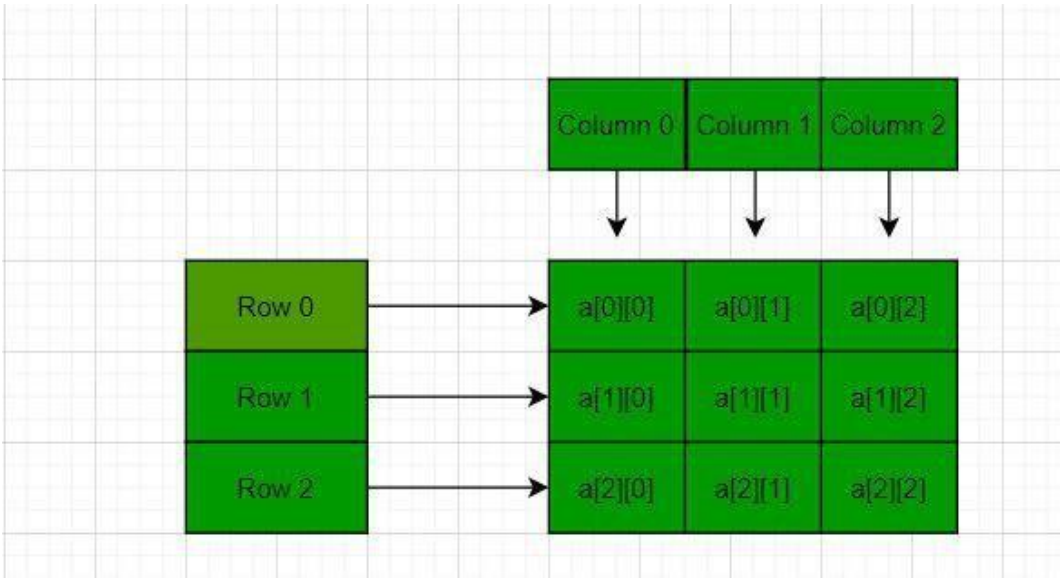
```
Value of variable :- 169
Value of variable using single pointer :- 169
```

Value of variable using double pointer :- 169

2-Dimensional Array:

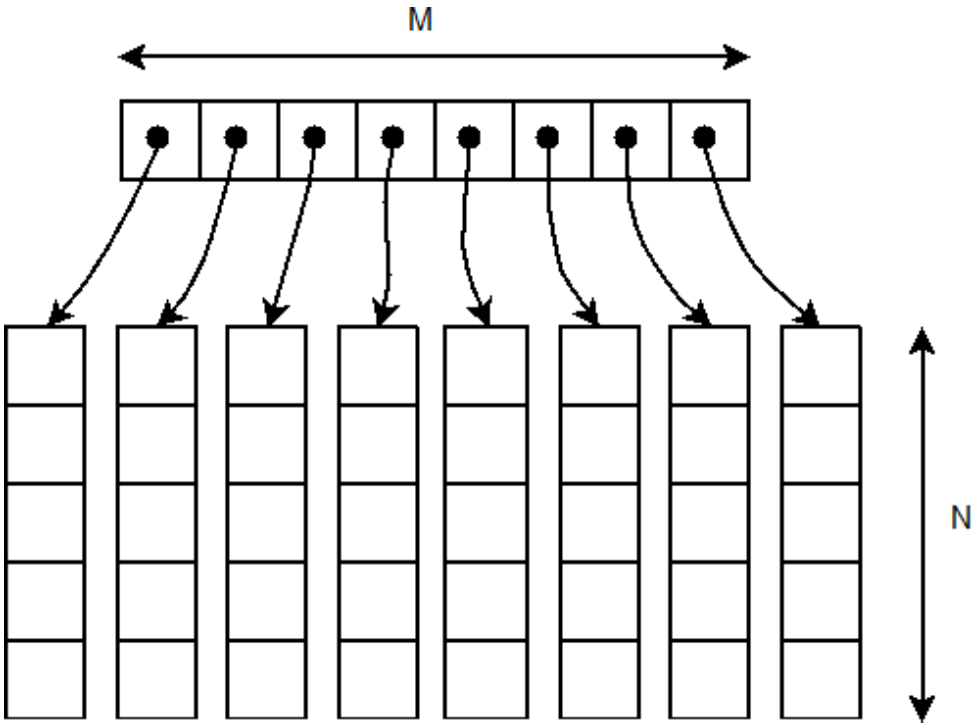
In C/C++, multidimensional arrays in simple words as an array of arrays. Data in multidimensional arrays are stored in tabular form (in row major order).

Below is the diagrammatic representation of 2D arrays:



Using array of Pointers:

We can dynamic ally create an array of pointers of size M and then dynamic ally allocate memory of size N for each row, as shown below:



Example:

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

int main() {

   // Create an array of 4 int* (pointers to int)
   // This is the first level of dynamic allocation
   int ** p = new int * [4];
```

```
// For each row (4 rows), allocate a dynamic array of 3 integers
// This makes it a 2D array (4x3) using double pointers
for(int i = 0; i < 4; i++) {
    p[i] = new int[3];
}

// Input values into the 2D array
for(int i = 0; i < 4; i++) {
    int * a = p[i];          // a points to the current row
    for(int j = 0 ; j < 3; j++) {
        cin >> *(a + j);     // store input at p[i][j]
    }
}

// Display the values of the 2D array
for(int i = 0; i < 4; i++) {
    int * a = p[i];
    cout << "Row " << i << ": ";
    for(int j = 0 ; j < 3; j++) {
        cout << *(a + j) << " ";   // print p[i][j]
    }
    cout << endl;
}

// Free memory for each row (3 integers per row)
for (int i = 0; i < 4; i++) {
    delete [] p[i];
}
```

```
    // Free memory for the array of row pointers
    delete [] p;


    return 0;
}
```