CODESCRACKER

- C++ Octal to Decimal
- C++ Octal to Binary
- C++ Octal to Hexadecimal
- C++ Hexadecimal to Decimal
- C++ Hexadecimal to Binary
- C++ Hexadecimal to Octal
- C++ Pattern Programs
- C++ Pattern Programs
- C++ Print Diamond Pattern
- C++ Print Floyd's Triangle
- C++ Print Pascal's Triangle
- C++ Array Programs
- C++ 1D Array Program
- C++ Linear Search
- C++ Binary Search
- Find Largest Element in Array
- Find Smallest Element in Array
- Find Second Largest Element
- Find Second Smallest Element
- C++ Sum of All Elements
- C++ Multiply All Elements
- C++ Element on Even Position
- C++ Element on Odd Position
- C++ Print Even Numbers in Array
- C++ Print Odd Numbers in Array
- C++ Count Even/Odd Numbers
- C++ Sum of Even/Odd Numbers
- C++ Count Positive Negative Zero
- C++ Reverse an Array
- C++ Insert Element in Array
- C++ Delete Element from Array
- C++ Merge two Arrays
- C++ Bubble Sort
- C++ Selection Sort
- C++ Insertion Sort
- C++ Common Elements
- C++ 2D Array Programs
- C++ Add Two Matrices
- C++ Subtract Two Matrices
- C++ Transpose Matrix
- C++ Multiply Two Matrices
- C++ 3D Array Programs
- C++ String Programs
- C++ Print String
- C++ Find Length of String
- C++ Compare Two Strings
- C++ Copy String
- C++ Concatenate String
- C++ Reverse a String
- C++ Delete Vowels from String
- C++ Delete Word from String
- C++ Count Character in String
- C++ Count Word in String
- C++ Frequency of Word
- C++ Remove Spaces from String
- C++ Sort a String
- C++ Uppercase to Lowercase
- C++ Lowercase to Uppercase
- C++ Swap Two Strings
- C++ Check Anagram or Not

C++ Three Dimensional Array Programs

In this article, you will learn and get code to implement three dimensional (3D) array in C++. Here are the list of programs available in this article:

- Initialize and Print Three Dimensional (3D) Array
- Print Three Dimensional Array with Index

- Program to show, how to access elements of Three Dimensional Array ?
- Receive dimension and elements for 3D array from user and print the array

Note – A Three Dimensional (3D) array comes in the category of multi-dimensional array. Multi-dimensional array means array of arrays.

Note – A Three Dimensional (3D) array is a collection of Two Dimensional (2D) arrays.

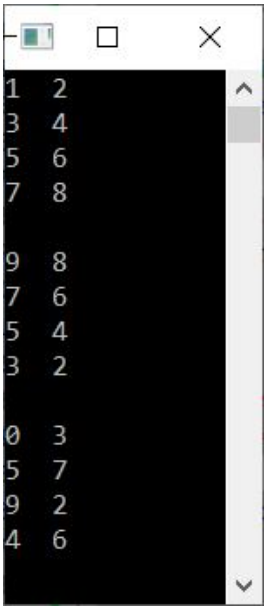
Note – In 3D array, there are three dimensions (subscripts). The first shows block size, second for row size, and the third is for column size. Here row and column size is the dimension of 2D array. Whereas the block size shows how many numbers of 2D arrays. For example, if the dimension of 3D array is **3*4*2.** Means 3 2D array of 4*2 dimension. Further means 3 2D array of size 4 rows and 2 columns.

3D Array Program in C++

This program initializes elements to three dimensional array named **threeDimArr[][][]** of size **3*4*2.** After all the 24 elements gets initialized to it, we've printed the 3D array back on the output screen as shown in the program given below:

```
#include<iostream>
using namespace std;
int main()
{
    int i, j, k;
    int threeDimArr[3][4][2] = {
        { {1, 2}, {3, 4}, {5, 6}, {7, 8} },
        { {9, 8}, {7, 6}, {5, 4}, {3, 2} },
        { {0, 3}, {5, 7}, {9, 2}, {4, 6} }
        };
    for(i=0; i<3; i++)
    {
        for(j=0; j<4; j++)
        {
            for(k=0; k<2; k++)
                cout<<threeDimArr[i][j][k]<<"  ";
            cout<<endl;
        }
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

This program was build and run under *Code::Blocks* IDE. Here is its sample output:



Note – As you can see, there are three 2-dimensional array of size 4*2 each. That is, each two dimensional array contains 4 rows and 2 columns.

To print 3D array, you have to use three **for** loops. Third **for** loop (the innermost loop) forms 1D array, Second **for** loop forms 2D array and the third **for** loop (the outermost loop) forms 3D array.

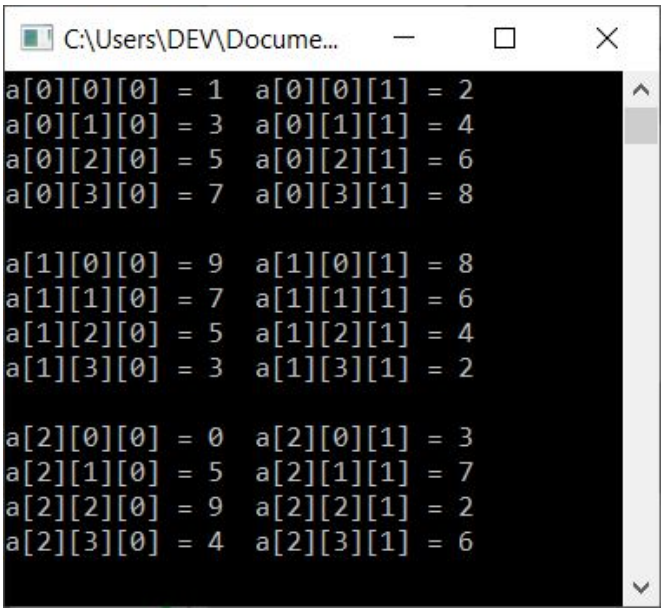
In other word, the outermost (first) *for loop* is for block size, the second is for row size of 2D array, and the third is for column size of 2D array.

Print 3D Array with Index

This program is similar to the previous program. In addition to the job done by previous program, this program prints all elements of 3D array along with indexes. So this program also shows how elements gets stored in 3D array:

```
#include<iostream>
using namespace std;
int main()
{
    int i, j, k;
    int a[3][4][2] = {
        { {1, 2}, {3, 4}, {5, 6}, {7, 8} },
        { {9, 8}, {7, 6}, {5, 4}, {3, 2} },
        { {0, 3}, {5, 7}, {9, 2}, {4, 6} }
        };
    for(i=0; i<3; i++)
    {
        for(j=0; j<4; j++)
        {
            for(k=0; k<2; k++)
                cout<<"a["<<i<<"]["<<j<<"]["<<k<<"] = "<<a[i][j][k]<<"  ";
            cout<<endl;
        }
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

The snapshot given below shows the sample output produced by this C++ program on three dimensional array:

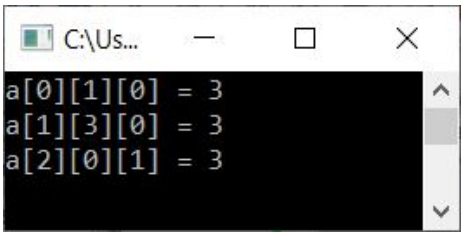


Access and Print Element of Three Dimensional Array

Now this C++ program shows how the element of a 3D array gets accessed. In this program, we've accessed three elements of 3D array with its index.

```
#include<iostream>
using namespace std;
int main()
{
    int i, j, k;
    int a[3][4][2] = {
        { {1, 2}, {3, 4}, {5, 6}, {7, 8} },
        { {9, 8}, {7, 6}, {5, 4}, {3, 2} },
        { {0, 3}, {5, 7}, {9, 2}, {4, 6} }
        };
    cout<<"a[0][1][0] = "<<a[0][1][0]<<endl;
    cout<<"a[1][3][0] = "<<a[1][3][0]<<endl;
    cout<<"a[2][0][1] = "<<a[2][0][1]<<endl;
    cout<<endl;
    return 0;
}
```

Here is its sample output:



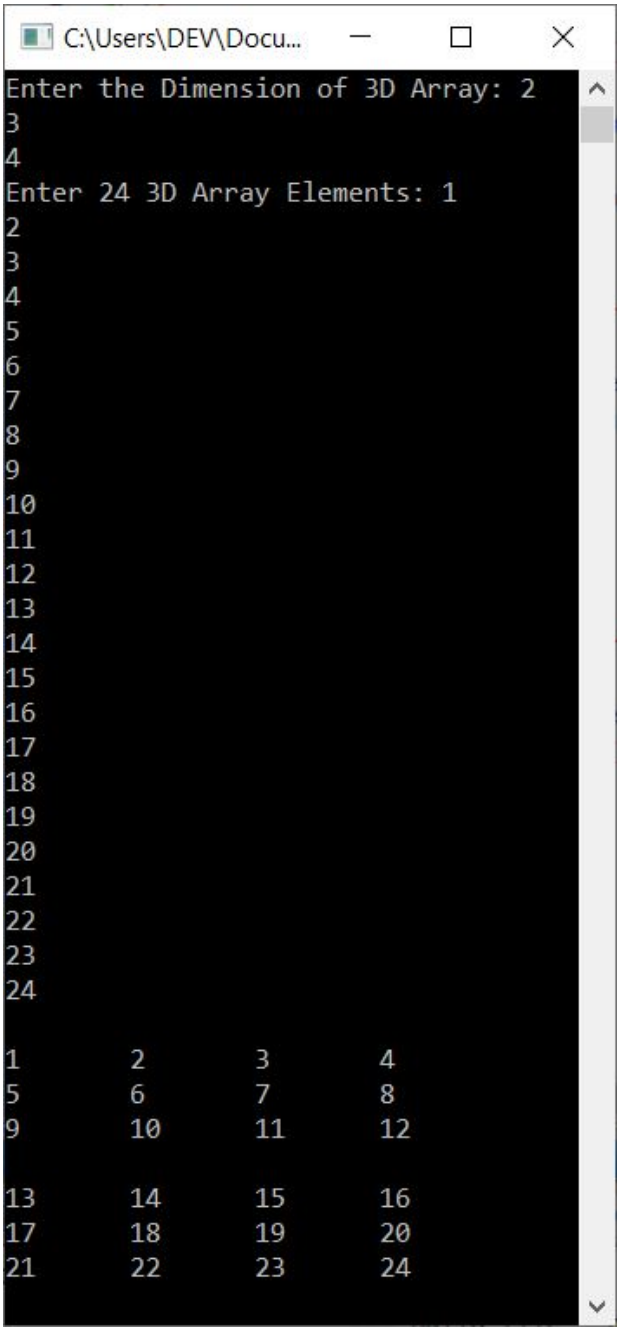
Note - In above program, a 1D array of two elements is constructed first. Then four such 1D arrays are placed one below the other to give a 2D array containing four rows. Then, three such 2D arrays are placed one behind the other to yield a 3D array containing three 2D arrays.

Receive Elements of 3D Array from User

This program allows user to enter the dimension and then elements for 3D array. Based on user inputs, we've printed the 3D array back on the output screen:

```
#include<iostream>
using namespace std;
int main()
{
    int i, j, k, a[10][10][10];
    int one, two, three;
    cout<<"Enter the Dimension of 3D Array: ";
    cin>>one>>two>>three;
    cout<<"Enter "<<one*two*three<<" 3D Array Elements: ";
    for(i=0; i<one; i++)
    {
        for(j=0; j<two; j++)
        {
            for(k=0; k<three; k++)
                cin>>a[i][j][k];
        }
    }
    cout<<endl;
    for(i=0; i<one; i++)
    {
        for(j=0; j<two; j++)
        {
            for(k=0; k<three; k++)
                cout<<a[i][j][k]<<"  ";
            cout<<endl;
        }
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

Here is its sample run with user input, **2, 3, 4** as dimensions and **1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24** as 24 3D array elements:



Same Program in Other Languages

- C Three-Dimensional Array Program
- Java Three-Dimensional Array Program

C++ Online Test

« Previous Program Next Program »

CodesCracker