



(<https://www.educba.com/software-development/>)



(<https://www.educba.com/2-d-arrays-in-c/>)

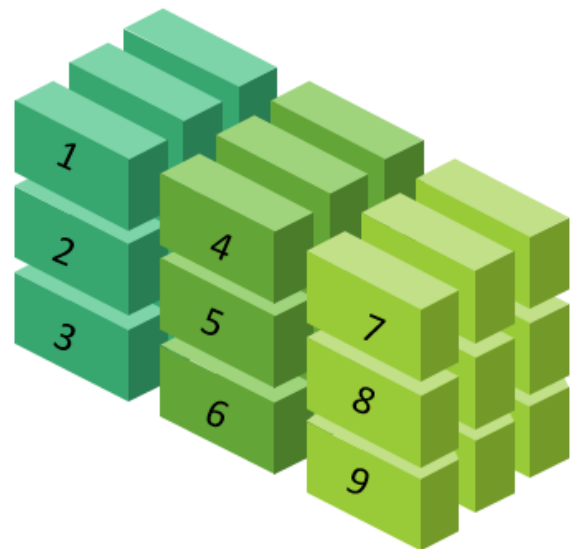


(<https://www.educba.com/multidimensional-array-in-c/>)

3-D Arrays in C



	Columns			
Rows	c[0][0]	c[0][1]	c[0][2]	c[0][3]
	c[1][0]	c[1][1]	c[1][2]	c[1][3]
	c[2][0]	c[2][1]	c[2][2]	c[2][3]



educba.com

Introduction to 3D Arrays in C

An Array is a group of elements with the same (homogeneous) data type. It is also called a Derived data type. As already noticed, a 3D array increases the space exponentially, and, an extra position added to locate the element in the array. In this topic, we are going to learn





(<https://www.educba.com/software-development/>)

search for? This 3D array is just for storing more amounts of data and representing the positions.

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

How can we define and implement them? Going further, let's understand those concepts.

Syntax:

In C, Dimensional arrays can be declared as follows:

```
Datatype variablename[size1][size2][size3]...[sizen];
```

So, in the same way, we can declare the 3-D array as:

```
int c[2][3][4];
```

The meaning of the above representation can be understood as:

1. The memory allocated to variable c is of data type int.
2. The total capacity that this array can hold is $2 \times 3 \times 4$, which is equal to 24 elements.
3. The data is being represented in the form of 2 arrays with 3 rows and 4 columns each.

		Columns			
c[0] Array	Rows	c[0][0]	c[0][1]	c[0][2]	c[0][3]





(<https://www.educba.com/software-development/>)

		Columns			
c[1] Array	Rows	c[0][0]	c[0][1]	c[0][2]	c[0][3]
		c[1][0]	c[1][1]	c[1][2]	c[1][3]
		c[2][0]	c[2][1]	c[2][2]	c[2][3]

The data inside the array can be accessed through the above representation. In 3D arrays representation (<https://www.educba.com/3d-arrays-in-c-plus-plus/>), the first square bracket represents the level of the array that has to be considered, second would be the number of rows, and the third one is for the number of columns.

🔗 Popular Course in this category



C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,612 ratings)

Course Price

\$79 ~~\$399~~

[View Course](#)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)





(<https://www.educba.com/software-development/>)

The index representation of the array for the first element always starts with zero and ends with size-1. So, for example, if the number of rows is 3, then the index representation for accessing the data in rows will be 0, 1 and 2. The same logic applies for the array level and column indexes too. For the above representation, to get the data of 1st level of the array with 2nd row 3rd column, we can access by `c[0][1][2]`.

Initializing 3D Arrays in C

We can initialize a 3D array similar to the 2-D array.

```
int c[2][4][3] = {
    {{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 mentioned above, the total number of elements that can be fit into the array would be `arraysize1*arraysize2*arraysize3`. Here it is `2*4*3`, which gives 24.

Inserting elements:

Similar to the 2D array, for inserting elements in a 3-D array, we need to insert the data in levels, rows, and columns. So, for this, we use the concept of loops. In the above process for initializing the data in the array, we had predefined the values.

Here, elements can be dynamically inserted by the user, as per the requirements. Below example code for inserting the elements.



Code



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
int c[2][4][3];
int i,j,k;
printf("Enter elements into 3-D array: ");
for(i=0;i<2;i++)
{
for(j=0;j<4;j++)
{
for(k=0;k<3;k++)
{
scanf("%d",&c[i][j][k]);
}
}
}
}
```

As observed in the code:

1. First, we are declaring the array variable and the dimensions of the array with the number of levels of the array, rows, and columns.
2. We are then declaring three variables for iterating over the elements in the array.
3. Then, for loops are used. The first loop is for the levels iteration, the second is for the rows and the third loop is for the columns.
4. Scanf function is used to read the data as we input, and then place the value inserted at those positions of i, j and k.





(<https://www.educba.com/software-development/>)

```
4
5
6
7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

As we have not used the printf function to display the output, the program written had only read the user inputted values. After writing the print function (using for loops), the output would be displayed as:

```
Enter elements into 3-D array1
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
      1      2      3
      4      5      6
      7      8      9
     10     11     12

     13     14     15
     16     17     18
     19     20     21
     22     23     24
```

Update Elements:

The updating of elements in the array can be done by either specifying a particular element to be replaced or by identifying a position where the replacement has to be done. For updating, we generally require the following details.

1. Elements of an array
2. Position/element, where it has to be inserted
3. The value to be inserted.



For updating the data in an array through element details, first, we need to search for that



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Firstly, let us go through an example where the position of the element to be updated is already known.

Code

```
#include <stdio.h>

int main()
{
    int c[2][4][3];
    int i,j,k,num;
    printf("Enter elements into 3-D array: ");
    for(i=0;i<2;i++)
    {
        for(j=0;j<4;j++)
        {
            for(k=0;k<3;k++)
            {
                scanf("%d",&c[i][j][k]);
            }
        }
    }

    c[1][1][1] = 85;
    for(i=0;i<2;i++)
    {
        for(j=0;j<4;j++)
        {
```





(<https://www.educba.com/software-development/>)

```
printf("%d", arr[0][0][0]);
}
printf("\n");
}
return 0;
}
```

In the above program, the element on the 1st level, 1st row, and 1st column is selected and the value of the data in that position has been updated.

Output for above is as follows:

```
Enter elements into 3-D array1
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
    1      2      3
    4      5      6
    7      8      9
   10     11     12

   13     14     15
   16     85     18
   19     20     21
   22     23     24
```

In the second example, we are going to show how the position of the element can be dynamically taken as a user inputted value and update the value of the element at that particular position.





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
{
int c[2][4][3];
int i,j,k,num;
printf("Enter elements into 3-D array: ");
for(i=0;i<2;i++)
{
for(j=0;j<4;j++)
{
for(k=0;k<3;k++)
{
scanf("%d",&c[i][j][k]);
}
}
}
printf("Enter the level , row and column number: ");
scanf("%d %d %d", &i,&j,&k);
printf("Enter the new number you want to update with: ");
scanf("%d" , &num);
c[i][j][k] = num;
for(i=0;i<2;i++)
{
for(j=0;j<4;j++)
{
for(k=0;k<3;k++)
{
```





(<https://www.educba.com/software-development/>)

```
printf("\n");
}
return 0;
}
```

The output is as follows. Here, we used the scanf function to read the value given by the user as per their choice for the position of an element based on array level, row, and column numbers.

```
Enter elements into 3-D array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Enter the level , row and column number: 0 1 2
Enter the new number you want to update with: 58
```

1	2	3
4	5	58
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24

As an exercise, can you try writing a program in updating the whole column of the matrix with user-inputted values?

Now, as we know, in the 3D array, we declare the size of the array at the beginning itself. We are aware of the size of the array but what if the user gives a random row and column number outside our array size?

```
Enter elements into 3-D array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Enter the level , row and column number: 2 2 2
Enter the new number you want to update with: 68
```

1	2	3
4	5	6
7	8	9





(<https://www.educba.com/software-development/>)

What if, we input more elements than required inside the array?

```
Enter elements into 3-D array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Enter the level , row and column number: Enter the new number you want to update with:
    4      5      6
    7      8      9
   10     11     12

   13     14     15
   16     17     18
   19     20     21
   22     23     24
```

Notice that as we had not written any if/else condition or try/catch blocks, the output of the matrix does not change. However, we can write the code using the above-mentioned conditions to display errors for such cases.

As the last example, are you not curious about, what happens if we skip some elements in between? What does my program do?

```
Enter elements into 3-D array: 1 2 3 4 6 7 8 9 11 12 13 14 15 17 18 19 20 21
25
26
27
28
Enter the level , row and column number: 1 2 3
```





(<https://www.educba.com/software-development/>)

```

19      20      21
23      24      25
26      27      78

```

As observed in the above output:

1. We missed up 4 values in the input, just by giving space and pressed enter
2. But we had got that scope to enter the 4 remaining elements.
3. Then we specified the last level, last row, and last column element to be changed to 78.

And the output is as expected isn't it?

Deleting Elements

After the concepts of insertion and updating of the data inside the 3D array, let's now see how we can delete an entire row from the array.

We have written a program in a simple format so that the concept of different operations can be understood easily.

Code:

```

#include <stdio.h>

int main()
{
    int c[2][4][3],i,j,k,num,x;
    printf("Enter elements into 3-D array: ");
    for(i=0;i<2;i++)
    {
        for(j=0;j<4;j++)

```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```

,
}
}

printf("Enter the value of row number to delete: ");
scanf("%d", &x);
for(i=0;i<2;i++)
{
for(j=0;j<4;j++)
{
if(j==x)
{
for(k=0;k<3;k++)
{
if((j+1)<4)
{
printf("\t%d" , c[i][j+1][k]);
}
}
}
}
}
}
j++;
}

```

```

}
j++;
}

else
{
for(k=0;k<3;k++)

```





(<https://www.educba.com/software-development/>)

```
printf("%d",  
}  
printf("\n");  
}  
}
```

Took the values of an array dynamically. The steps followed are:

1. Asked user to input the number (index) of the row that has to be deleted.
2. Using for loop iteration of array levels, rows, and columns. We are comparing if the row number and the user input number are matching or not.
3. If they are matching and if the row number is less than the size of the array, we are printing the next row. Else, we are printing the row as it is.
4. Here, as we didn't have any condition on the level of the array, the row number specified is deleted from both the array levels.

The output is as follows:

```
Enter elements into 3-D array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Enter the value of row number to delete: 1
    1      2      3
    7      8      9
   10     11     12

   13     14     15
   19     20     21
   22     23     24

Enter elements into 3-D array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Enter the value of row number to delete: 3
    1      2      3
    4      5      6
    7      8      9
```





(<https://www.educba.com/software-development/>)

```
Enter the value of row number to delete: 0
```

```

4      5      6
7      8      9
10     11     12

16     17     18
19     20     21
22     23     24
```

What if, we give the row number outside the array boundary?

```
Enter elements into 3-D array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Enter the value of row number to delete: 4
```

```

1      2      3
4      5      6
7      8      9
10     11     12

13     14     15
16     17     18
19     20     21
22     23     24
```

It will not find the row to delete and exit the program by printing the whole array.

As already known, we can even declare the values for the number of rows and columns dynamically and write the program accordingly.

Doesn't this look simple and easy to learn?

As an exercise, can you try in deleting a particular element for the 3d array now?

Conclusion

In this section, we have learned the basic operations on 3-dimensional arrays.





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Try solving the basic operations of the 3d arrays and have fun learning C.

Recommended Articles

This is a guide to 3D Arrays in C. Here we discuss the initializing of a 3D array similar to the 2D array and elements of Array. You may also look at the following article to learn more –

1. [Arrays in R \(https://www.educba.com/arrays-in-r/\)](https://www.educba.com/arrays-in-r/)
2. [Arrays in C# \(https://www.educba.com/arrays-in-c-sharp/\)](https://www.educba.com/arrays-in-c-sharp/)
3. [3D Arrays in Java \(https://www.educba.com/3d-arrays-in-java/\)](https://www.educba.com/3d-arrays-in-java/)
4. [Arrays in PHP \(https://www.educba.com/arrays-in-php/\)](https://www.educba.com/arrays-in-php/)

C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

- ☒ 3 Online Courses
- ☒ 5 Hands-on Projects
- ☒ 34+ Hours
- ☒ Verifiable Certificate of Completion
- ☒ Lifetime Access

Learn More

<https://www.educba.com/software-development/courses/c-programming-course/?btnz=educba-inline-banner3>





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

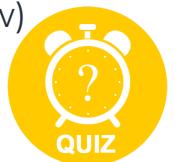
Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

development-tutorials/?source=footer)

Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)

Software Development Course - All in One Bundle
(<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

