Coding Deletion Operation in Array Using C Language (With Notes)

codewithharry.com/videos/data-structures-and-algorithms-in-hindi-11

In the last tutorial, we had learned about the first two primary operations in an array ADT, traversal and insertion. Today, we will study the third one, *deletion*.

Programming a deletion differs very slightly from programming an insertion. In insertion, we had to shift elements to their adjacent right to create a void at the desired place to insert a new element, but in deletion, we'll shift the elements to their adjacent left to fill the void created after deleting an element at some index.

Let us now code this out or rather transform the code we constructed to insert the element. I have attached the snippet below.

Understanding code snippet 1:

1. One thing which will remain as it is, is the display function.
2. We have to make minimal changes in the insertion function to make it a deletion function. Rename it *indDeletion*. The index and the array, and its size will be our only parameters this time.
3. Replace the right shift with the left shift. Just assign array[i], the value present in array[i+1].
4. And we are done deleting the element at some specified index.

```
#include <stdio.h>

void display(int arr[], int n)
{
    // Code for Traversal
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void indDeletion(int arr[], int size, int index)
{
    // code for Deletion
    for (int i = index; i < size-1; i++)
    {
        arr[i] = arr[i + 1];
    }
}

int main()
{
    int arr[100] = {7, 8, 12, 27, 88};
    int size = 5, element = 45, index = 0;
    display(arr, size);
    indDeletion(arr, size, index);
    size -= 1;
    display(arr, size);
    return 0;
}
```

Code Snippet 1: Deletion in an array:

We can now check if the program actually works for deleting the element at some index. We'll create an array with 5 elements and display it before and after deleting an element at index 0.

Refer to the output below:

```
7 8 12 27 88
8 12 27 88
PS D:\MyData\Business\code playground\Ds & Algo with Notes\Code>
```

Figure 1: Output of the above program

So, the code works fine. Element at index 0 got deleted, and the rest of the elements shifted left to fill the void created after deletion. And this was all about the deletion. Only a subtle change of code helped transform insertion into deletion. Practice these operations on your desktop till you feel confident about them.

Thank you for being with me throughout. I hope you enjoyed the tutorial. If you appreciate my work, please let your friends know about this course too. If you haven't checked out the whole playlist yet, move on to codewithharry.com or my YouTube

channel to access it. See you in the next tutorial, where we'll learn to code both the search methods we have studied, linear and binary, in C language. Till then, keep learning.