



## Challenge 2: Delete an Element at a Specific Index

Test your knowledge by solving a challenge in this lesson.

We'll cover the following



- Problem statement
  - Sample input
  - Sample output
- Coding exercise

### Problem statement#

Your task is to write a function `delete_element`. In its input parameters, the function will take:

- Pointer `arr` of type `int` by reference. Pointer `arr` is pointing to dynamically allocated array;
- Variables `size` and `index` of type `int`.

```
void delete_element ( int *&arr , int size , int index )
```

Your task is to delete the element in an array at the given `index`.



(/learn)

**Note:** By default, pointers are passed by value. This means that if we point the pointer to a different memory location inside the function, it still points to the previous memory location outside the function. Here, we are passing pointers by reference, so if we change the address stored in a pointer inside the function, it is also changed outside the function.

## Sample input#

```
delete_element ({10, 67, 98, 31}, 4 , 2 );
```

## Sample output#

```
{10, 67, 31}
```

## Coding exercise#

Before diving directly into the solution, try to solve it yourself. Then check if your code passes all the test cases.




Please don't modify the given function `delete_element`. Otherwise, your code will not compile.


Good luck! 🍀


```
1 void delete_element(int * & arr, int size, int index) {  
2     // Write your code here
```





3 }

















🎉 If you have solved the problem, congratulations!

In case you are stuck, go over the solution review in the next lesson.


 Back

Next 

Solution Review: Set the Odd Element...

Solution Review: Delete an Element at...

☒ Mark as Completed

 Report an Issue