Solution Review: Set the Odd Elements in a Dynamic Array to -1

Let's see the detailed solution review of the challenge given in the previous lesson.

We'll cover the following

- Solution
- Explanation
  - set_odd function

Solution

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

// printArray function
void printArray(int * arr, int size) {
  for (int i = 0; i < size; i++) {
    cout << arr[i] << " ";
  }
  cout << endl;
}

// set_odd function
void set_odd(int * arr, int size) {
  // Traverse array
  for (int i = 0; i < size; i++) {
    // Check if current element is odd
    if (arr[i] % 2 != 0) {
```

```
19        // Set odd element to -1
20        arr[i] = -1;
21      }
22    }
23  }
24
25  // main function
26  int main() {
27    // Initialize size of an array
28    int size = 5;
```

Explanation#

set_odd function

The `set_odd` function is of type `void`. It takes a pointer `arr` that points to an array of type `int` and `size` of an array in its input parameters.

If a number is not divisible by 2, it is odd. Traverse the array `arr[]`. If any element is not divisible by 2, set its value to `-1`.

In the upcoming lesson, we will solve a slightly more difficult challenge related to dynamic allocation.

Back

Challenge 1: Set the Odd Elements in ...

Next

Challenge 2: Delete an Element at a S...

Mark as Completed

Report an Issue