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

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) {
      // Set odd element to -1
      arr[i] = -1;
    }
  }
}

// main function
int main() {
  // Initialize size of an array
  int size = 5;
  // Declare dynamic array
  int * arr = new int[size];
  // Initialize array
  for (int i = 0; i < size; i++) {
    arr[i] = i;
  }
  // Call printArray function

  printArray(arr, size);
  // Call set_odd function
  set_odd(arr, size);
  // Call printArray function
  printArray(arr, size);
  return 0;
}
```

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.