

**Name :- Ahmed Ali Asif**

**SAP ID :- 55346**

**Programme :- BSCS 3-1**

### **Assignment #1**

**Q1**

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int numStudents = 10; // Not using a constant
```

```
    int studentAges[10]; // Hardcoded size
```

```
    cout << "Please input the ages for 10 students." << endl;
```

```
    // Getting ages from the user
```

```
    for (int i = 0; i < numStudents; i++) {
```

```
        cout << "Enter age for student " << (i + 1) << ": ";
```

```
        cin >> studentAges[i];
```

```
    }
```

```
    // Finding the largest age
```

```
    int maxAge = 0; // Initializing to 0, might not work for negative ages
```

```
    for (int j = 0; j < numStudents; j++) {
```

```
        if (studentAges[j] > maxAge) {
```

```
            maxAge = studentAges[j]; // Finding the maximum age
```

```

    }
}

```

**// Outputting the largest age**

```
cout << "The maximum age is: " << maxAge << endl;
```

```
return 0;
```

```
}
```

The screenshot shows a web browser window with the URL `programiz.com/cpp-programming/online-compiler/`. The page features a header with the Programiz logo and a banner for "Premium Coding Courses by Programiz". Below the header, the main content area is divided into two panels. The left panel, titled "main.cpp", contains the following C++ code:

```

11  cout << "Enter age for student " << (i + 1) << ": ";
12  cin >> studentAges[i];
13  }
14
15  // Finding the largest age
16  int maxAge = 0; // Initializing to 0, might not work for
17  // negative ages
18  for (int j = 0; j < numStudents; j++) {
19      if (studentAges[j] > maxAge) {
20          maxAge = studentAges[j]; // Finding the maximum age
21      }
22
23  // Outputting the largest age
24  cout << "The maximum age is: " << maxAge << endl;
25
26  return 0;
27 }
28

```

The right panel, titled "Output", shows the execution results:

```

/tmp/nfbzVJFVUL.o
Please input the ages for 10 students.
Enter age for student 1: 11
Enter age for student 2: 12
Enter age for student 3: 14
Enter age for student 4: 15
Enter age for student 5: 8
Enter age for student 6: 18
Enter age for student 7: 17
Enter age for student 8: 10
Enter age for student 9: 19
Enter age for student 10: 21
The maximum age is: 21

=== Code Execution Successful ===

```

The browser's taskbar at the bottom shows the time as 11:08 PM on 9/20/2024.

**Q2**

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
int n; // using a vague variable name  
cout << "How many elements for the arrays? ";  
cin >> n;  
  
// Using dynamic memory allocation, but could be avoided  
int *arr1 = new int[n];  
int *arr2 = new int[n];  
int *arr3 = new int[n];  
int *result = new int[n];  
  
// Filling array 1  
cout << "Input values for Array 1:" << endl;  
for (int idx = 0; idx < n; idx++) {  
    cout << "Enter number for element " << idx << ": "; // Unnecessary  
repeated text  
    cin >> arr1[idx];  
}  
  
// Filling array 2  
cout << "Input values for Array 2:" << endl;  
for (int idx = 0; idx < n; idx++) {  
    cout << "Enter number for element " << idx << ": ";  
    cin >> arr2[idx];  
}
```

**// Filling array 3**

**cout << "Input values for Array 3:" << endl;**

**for (int idx = 0; idx < n; idx++) {**

**cout << "Enter number for element " << idx << ": ";**

**cin >> arr3[idx];**

**}**

**// Summing the arrays**

**for (int i = 0; i < n; i++) {**

**result[i] = arr1[i] + arr2[i] + arr3[i]; // Direct summation in one step**

**}**

**// Displaying the summed array**

**cout << "Here is the Sum Array:" << endl;**

**for (int j = 0; j < n; j++) {**

**cout << "Sum at index " << j << ": " << result[j] << endl; // Non-standard output format**

**}**

**// Not verifying memory allocation or using smart pointers**

**delete[] arr1;**

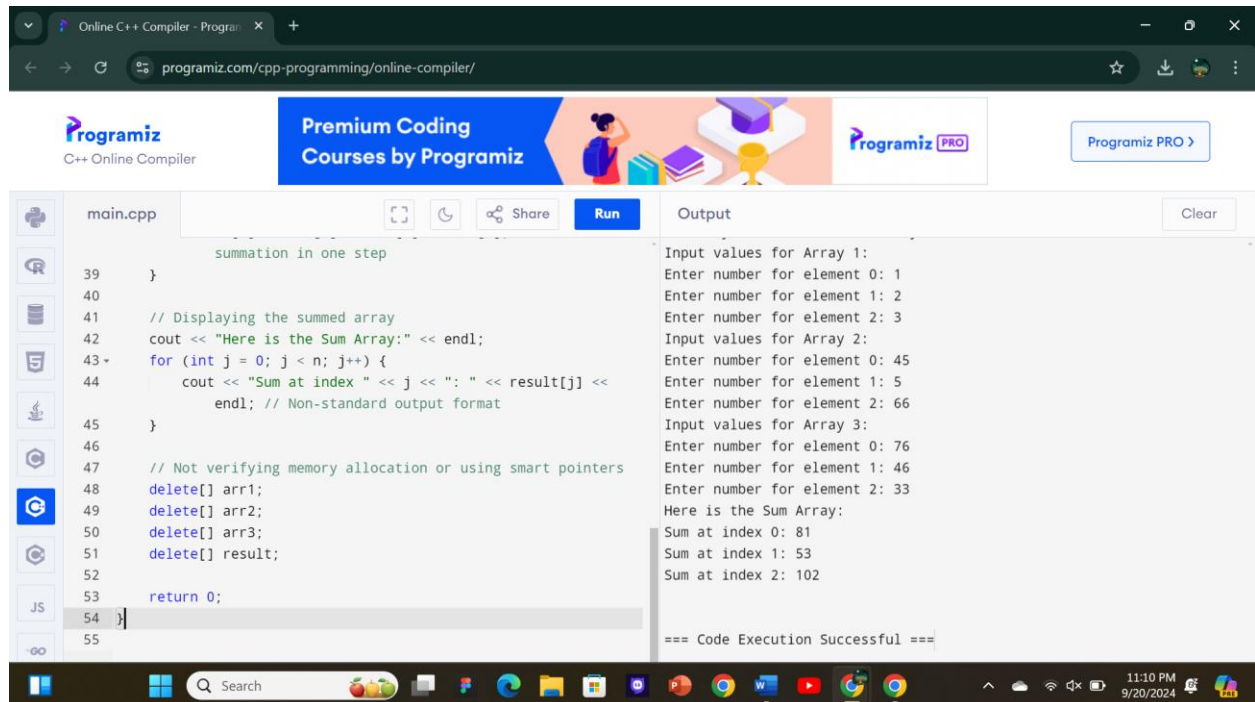
**delete[] arr2;**

**delete[] arr3;**

**delete[] result;**

**return 0;**

**}**



The screenshot shows the Programiz C++ Online Compiler interface. The code editor on the left contains a C++ program for summing three arrays. The output window on the right shows the program's execution, including prompts for array elements and the resulting sums at each index. The status bar at the bottom indicates "Code Execution Successful".

```
main.cpp
39     }
40
41     // Displaying the summed array
42     cout << "Here is the Sum Array:" << endl;
43     for (int j = 0; j < n; j++) {
44         cout << "Sum at index " << j << ": " << result[j] <<
            endl; // Non-standard output format
45     }
46
47     // Not verifying memory allocation or using smart pointers
48     delete[] arr1;
49     delete[] arr2;
50     delete[] arr3;
51     delete[] result;
52
53     return 0;
54 }
55
```

Output

```
Input values for Array 1:
Enter number for element 0: 1
Enter number for element 1: 2
Enter number for element 2: 3
Input values for Array 2:
Enter number for element 0: 45
Enter number for element 1: 5
Enter number for element 2: 66
Input values for Array 3:
Enter number for element 0: 76
Enter number for element 1: 46
Enter number for element 2: 33
Here is the Sum Array:
Sum at index 0: 81
Sum at index 1: 53
Sum at index 2: 102

=== Code Execution Successful ===
```

**Q3**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int lengthOfArray;**

**cout << "Please specify how big the array is: ";**

**cin >> lengthOfArray;**

**int\* dataArray = new int[lengthOfArray];**

**cout << "Now input the elements of the array one by one: " << endl;**

```
for (int index = 0; index < lengthOfArray; index++) {  
    cout << "Please enter value for element " << index << ": ";  
    cin >> dataArray[index];  
}
```

```
int itemToSearch;  
cout << "What value are you looking for? ";  
cin >> itemToSearch;
```

```
bool isFound = false;  
int location = -1;  
for (int x = 0; x < lengthOfArray; x++) {  
    if (dataArray[x] == itemToSearch) {  
        isFound = true;  
        location = x;  
        break;  
    }  
}
```

```
// Output result  
if (isFound) {  
    cout << "Success! Item found at array index: " << location << endl;  
// Informal language  
} else {
```

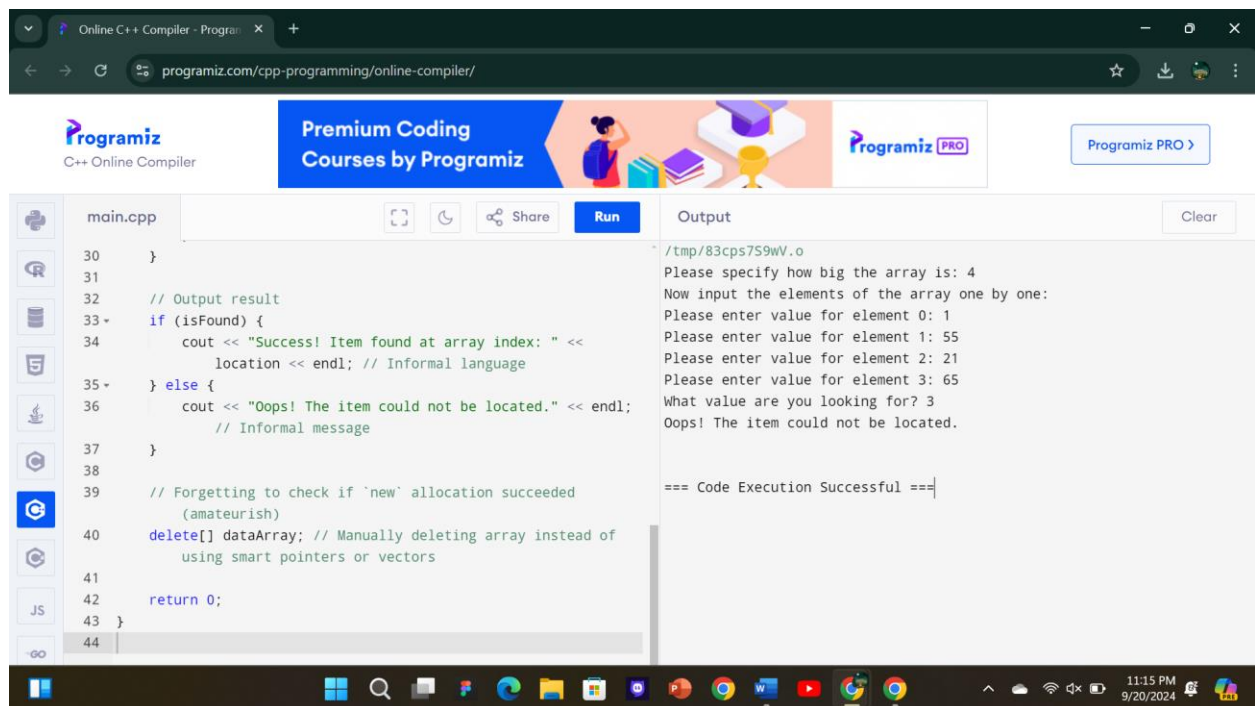
```
cout << "Oops! The item could not be located." << endl; // Informal message
```

```
}
```

```
delete[] dataArray; // Manually deleting array instead of using smart pointers or vectors
```

```
return 0;
```

```
}
```



The screenshot shows a web browser window with the URL `programiz.com/cpp-programming/online-compiler/`. The page features the Programiz logo and a banner for "Premium Coding Courses by Programiz". Below the banner is a code editor with a file named `main.cpp`. The code in the editor is as follows:

```
30 }
31
32 // Output result
33 if (isFound) {
34     cout << "Success! Item found at array index: " <<
        location << endl; // Informal language
35 } else {
36     cout << "Oops! The item could not be located." << endl;
        // Informal message
37 }
38
39 // Forgetting to check if 'new' allocation succeeded
    (amateurish)
40 delete[] dataArray; // Manually deleting array instead of
    using smart pointers or vectors
41
42 return 0;
43 }
44
```

To the right of the code editor is an "Output" panel. It shows the execution output for the program, which includes prompts for array size and elements, followed by the user's input and the program's response. The output is as follows:

```
/tmp/83cps7S9wV.o
Please specify how big the array is: 4
Now input the elements of the array one by one:
Please enter value for element 0: 1
Please enter value for element 1: 55
Please enter value for element 2: 21
Please enter value for element 3: 65
What value are you looking for? 3
Oops! The item could not be located.

=== Code Execution Successful ===
```

The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 11:15 PM on 9/20/2024.