C++ Assignments | Arrays - 2 | Week 5

1.Count the number of elements strictly greater than x.

2.WAP to find the largest three elements in the array.

3.Check if the given array is sorted or not

4.Find the difference between the sum of elements at even indices to the sum of elements at odd indices.

5.Given an array of integers, change the value of all odd indexed elements to its second multiple and increment all even indexed values by 10.

6.Find the unique number in a given Array where all the elements are being repeated twice with one value being unique.

7.If an array arr contains n elements, then check if the given array is a palindrome or not .

8.Find the error.

```
double getAverage(int arr[], int size);
int main () {
int balance[5] = {1000, 2, 3, 17, 50};
double avg;
avg = getAverage( balance[0], 5 ) ;
cout << "Average value is: " << avg << endl;
return 0;
}
```

Output:-

```
1.#include <iostream>
using namespace std;

int countGreaterThanX(int arr[], int size, int x) {
   int count = 0;
   for (int i = 0; i < size; i++) {
      if (arr[i] > x) {
         count++;
      }
   }
   return count;
}
```

```
int main() {
   int arr[] = {1, 4, 7, 3, 6, 8, 5};
   int x = 5;
   int size = sizeof(arr) / sizeof(arr[0]);
   cout << "Number of elements greater than " << x << " is: " <<
countGreaterThanX(arr, size, x) << endl;
   return 0;
}
```

——-(Number of elements greater than 5 is: 3)

2.

```
#include <iostream>
#include <climits>
using namespace std;

void findThreeLargest(int arr[], int size) {
   if (size < 3) {
      cout << "Array should have at least three elements." << endl;
      return;
   }

   int first = INT_MIN, second = INT_MIN, third = INT_MIN;
   for (int i = 0; i < size; i++) {
      if (arr[i] > first) {
         third = second;
         second = first;
         first = arr[i];
      } else if (arr[i] > second) {
         third = second;
         second = arr[i];
      } else if (arr[i] > third) {
         third = arr[i];
      }
   }
   cout << "The three largest elements are: " << first << ", " << second << ", "
<< third << endl;
}

int main() {
   int arr[] = {10, 4, 3, 50, 23, 90};
   int size = sizeof(arr) / sizeof(arr[0]);
   findThreeLargest(arr, size);
   return 0;
}
```

—The three largest elements are: 90, 50, 23

3.

```
#include <iostream>
using namespace std;
```

```
bool isSorted(int arr[], int size) {
   for (int i = 1; i < size; i++) {
      if (arr[i] < arr[i - 1]) {
         return false;
      }
   }
   return true;
}

int main() {
   int arr[] = {1, 2, 3, 4, 5};
   int size = sizeof(arr) / sizeof(arr[0]);
   if (isSorted(arr, size)) {
      cout << "The array is sorted." << endl;
   } else {
      cout << "The array is not sorted." << endl;
   }
   return 0;
}
——The array is sorted.
4.#include <iostream>
using namespace std;

int sumDifference(int arr[], int size) {
   int evenSum = 0, oddSum = 0;
   for (int i = 0; i < size; i++) {
      if (i % 2 == 0) {
         evenSum += arr[i];
      } else {
         oddSum += arr[i];
      }
   }
   return evenSum - oddSum;
}

int main() {
   int arr[] = {10, 4, 3, 50, 23, 90};
   int size = sizeof(arr) / sizeof(arr[0]);
   cout << "Difference between sums of even and odd indexed elements is: "
<< sumDifference(arr, size) << endl;
   return 0;
}
—-Difference between sums of even and odd indexed elements is: -24
5.#include <iostream>
using namespace std;

void modifyArray(int arr[], int size) {
```

```
    for (int i = 0; i < size; i++) {
        if (i % 2 == 0) {
            arr[i] += 10;
        } else {
            arr[i] *= 2;
        }
    }
}

int main() {
    int arr[] = {1, 4, 7, 3, 6, 8, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    modifyArray(arr, size);
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

——11 8 17 6 16 16 15

6.

```
#include <iostream>
using namespace std;

int findUnique(int arr[], int size) {
    int unique = 0;
    for (int i = 0; i < size; i++) {
        unique ^= arr[i];
    }
    return unique;
}

int main() {
    int arr[] = {2, 3, 5, 4, 5, 3, 4};
    int size = sizeof(arr) / sizeof(arr[0]);
    cout << "The unique element is: " << findUnique(arr, size) << endl;
    return 0;
}
```

—-The unique element is: 2

7.

```
#include <iostream>
using namespace std;

bool isPalindrome(int arr[], int size) {
    for (int i = 0; i < size / 2; i++) {
        if (arr[i] != arr[size - i - 1]) {
            return false;
        }
    }
    return true;
```

```
}

int main() {
    int arr[] = {1, 2, 3, 2, 1};
    int size = sizeof(arr) / sizeof(arr[0]);
    if (isPalindrome(arr, size)) {
        cout << "The array is a palindrome." << endl;
    } else {
        cout << "The array is not a palindrome." << endl;
    }
    return 0;
}
```

—-The array is a palindrome.

8.The key error was that `balance[0]` (which is the first element of the array) was being passed to `getAverage` instead of the entire array `balance`. This is corrected by passing `balance` directly.

So the correct code is:-

```
#include <iostream>
using namespace std;

double getAverage(int arr[], int size) {
    double sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    return sum / size;
}

int main () {
    int balance[5] = {1000, 2, 3, 17, 50};
    double avg;
    avg = getAverage(balance, 5); // Correct: passing balance
    cout << "Average value is: " << avg << endl;
    return 0;
}
```