EAST DELTA UNIVERSITY

Assignment: 02

Course title: Data Structure Laboratory

Course Code: CSE 212

Submitted by,

Name: Mohammad Fahim
ID: 242002112
Section: 6
Department: CSE

Submitted to,

Name: Mohammad Akbar Bin Shah

Designation: Lecturer, SoSET

Date Of Submission: 22-02-2025

Problem: 01
Arith Geo:
Source Code(c++):

```
#include <bits/stdc++.h>

using namespace std;

string ArithGeo(int arr[], int n)
{
    if (n < 2) return "-1";

    bool isArithmetic = true, isGeometric = true;
    int diff = arr[1] - arr[0];
    int ratio = arr[1] / arr[0];

    for (int i = 1; i < n; i++) {
        if (arr[i] - arr[i - 1] != diff)
        {
            isArithmetic = false;
        }
        if (arr[i - 1] != 0 && arr[i] / arr[i - 1] != ratio)
        {
            isGeometric = false;
        }
    }
```

```
    if (isArithmetic) return "Arithmetic";
    if (isGeometric) return "Geometric";
    return "-1";
}

int main()
{
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    int arr[n];
    cout << "Enter the elements: ";
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    cout << "Result: " << ArithGeo(arr, n) << endl;

    return 0;
}
```

Explanation:

The function determines whether a given sequence is arithmetic, geometric, or neither. It first calculates the common difference and ratio.Then, it iterates through the array to check if all elements follow the arithmetic or geometric sequence rules.

Input/Output:

```
Enter the number of elements: 4
Enter the elements: 5 10 15 20
Result: Arithmetic


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

```
Enter the number of elements: 4
Enter the elements: 2 4 8 16
Result: Geometric


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

```
Enter the number of elements: 4
Enter the elements: 2 8 19 32
Result: -1


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

Problem: 02

Remove Duplicates from Sorted Array:

Source Code(c++):

```
#include <iostream>

using namespace std;

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    if (n == 0) {
        cout << "0" << endl;
        return 0;
    }

    int nums[n];
    cout << "Enter sorted elements: ";
    for (int i = 0; i < n; i++) {
        cin >> nums[i];
    }

    int uniqueCount = 1;
    for (int i = 1; i < n; i++) {
        if (nums[i] != nums[i - 1]) {
            nums[uniqueCount++] = nums[i];
        }
```

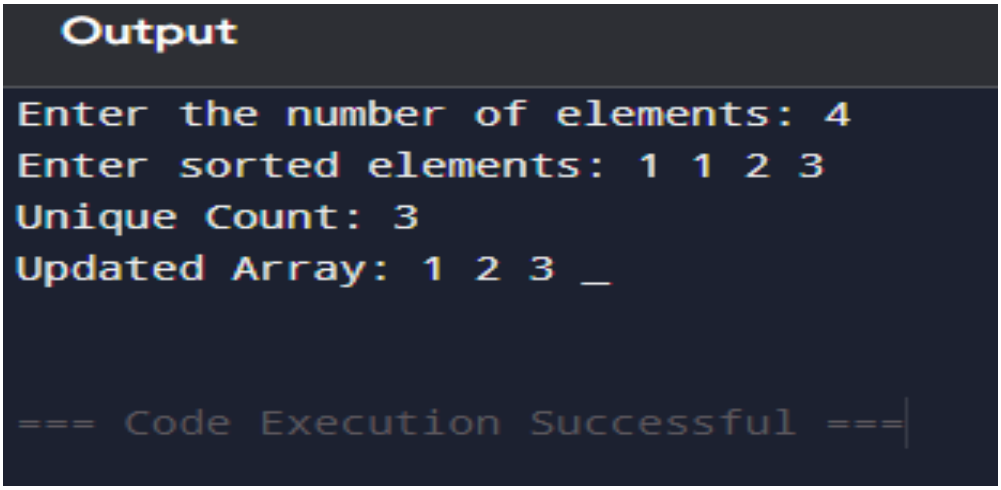
```
    }

    cout << "Unique Count: " << uniqueCount << endl;
    cout << "Updated Array: ";
    for (int i = 0; i < uniqueCount; i++) {
        cout << nums[i] << " ";
    }
    for (int i = uniqueCount; i < n; i++) {
        cout << "_ ";
    }
    cout << endl;
    return 0;
}
```

Explanation:

Since the array is already sorted, duplicates appear consecutively. The program uses a single traversal to shift unique elements to the front.It maintains a counter for unique values and modifies the array in place, saving space.

Input/Output:



Problem: 03

Add One to a Number Represented as an Array:

Source Code(c++):

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cout << "Enter number of digits: ";
    cin >> n;

    int arr[n];
    cout << "Enter digits: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    int carry = 1;
    for (int i = n - 1; i >= 0; i--)
    {
        arr[i] += carry;
        if (arr[i] < 10)
        {
            carry = 0;
            break;
        }
        arr[i] = 0;
```

```
    }

    if (carry)
    {
        cout << "Result: 1 ";
    }

    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }

    cout << endl;
    return 0;
}
```

Explanation:
It starts from the least significant digit and moves backward, updating digits accordingly. If an extra carry remains at the end, it prepends 1 to the output.

Input/Output: