FOP LAB ASSIGNMENT – I

Q1. Write a C++ program, take two strings as input from user and check if both strings are equal or not. If they are equal make them unequal by rotating string. e.g., Hello is turned into olleH etc.

```cpp
#include <iostream>
#include <cstring>

using namespace std;

int main() {

    string str1, str2;
    cout << "First String: ";
    cin >> str1;
    cout << endl;
    cout << "Second String: ";
    cin >> str2;

    if (str1 == str2)
    {
        cout << "Equal" << endl;
        char str2_[str2.length() + 1];
        for (int i = 0; i < str2.length(); i++)
        {
            str2_[i] = str2[str2.length() - 1 - i];
        }
        str2_[str2.length()] = '\0';

        cout << str1 << endl << str2_;
    }
    else
    {
        cout << "Unequal" << endl;
        cout << str1 << endl << str2;
    }

    return 0;
}
```

```
/tmp/9rM050MB90.o
First String: Nominal
Second String: Nominal
Equal
Nominal
lanimoN
```

Q2. Write a C++program for a string which may contain lowercase and uppercase characters. The task is to remove all duplicate characters from the string and find the resultant string.

```cpp
#include <iostream>
#include <cstring>

using namespace std;

int main() {

    int bin, count;
    string input;
    getline (cin, input);
    char result [input.length () + 1];

    for (int i = 0; i < input.length (); i++)
    {
        bin = 0;

        for (int j = 0; j <= i; j++)
        {
            if (input [i] == result [j])
            {
                bin = 1;
                break;
            }
        }

        if (bin == 0)
        {
            result [count] = input [i];
            count++;
        }
    }

    result [count] = '\0';
    cout << result;

    return 0;
}
```

```
/tmp/UncCcpU2mr.o
In magic, Abracadabra
In magic,Abrd
```

Q3. Suppose an integer array a[5] = {1,2,3,4,5}. Add more elements to it and display them in C++.

```cpp
#include <iostream>
#include <cstring>

using namespace std;

int main() {

    int array [5] = {1, 2, 3, 4, 5};

    array [5] = 8;
    array [6] = 9;

    for (int i = 0; i < 7; i++)
    {
        cout << array [i] << endl;
    }

    return 0;
}
```

```
/tmp/24fcALquw5.o
1
2
3
4
5
8
9
```

Q4. Write a C++ program that uses a while loop to find the largest prime number less
than a given positive integer N. Your program should take the value of N as input from the user and then find the largest prime number less than or equal to N. You

are not allowed to use any library or pre-existing functions to check for prime numbers.

```cpp
#include <iostream>

using namespace std;

int main() {

    int arr [6] = {2, 3, 5, 7, 11, 13};

    int largest_prime = 2;
    int n;
    cin >> n;

    for (int i = 2; i < n; i++)
    {
        for (int j = 0; j <= 5; j++)
        {
            if (arr [j] == i)
            {
                continue;
            }

            if (i % arr [j] == 0)
            {
                break;
            }

            if (j == 5)
            {
                b = i;
            }
        }
    }

    cout << b;

    return 0;
```

```
/tmp/5KOixdMtFY.o
78
73
```

Q5. Implement Bubble Sort on an array of 6 integers.

```cpp
#include <iostream>

using namespace std;

int main() {

    int arr [6] = {37, 30, 55, 23, 71, 44};
    int temp;

    for (int i = 0; i < 6; i++)
    {
        for (int j = 0; j< 6; j++)
        {
            if (arr [i] < arr [j])
            {
                temp = arr [i];
                arr [i] = arr [j];
                arr [j] = temp;
            }
        }
    }

    for (int i = 0; i < 6; i++)
    {
        cout << arr [i] << endl;
    }

    return 0;
}
```

```
/tmp/5KOixdMtFY.o
23
30
37
44
55
71
```

Q6. For an aircraft flying towards a location it has to drop a payload on, using as input the required values, determine the acceleration required such that if the payload were to be dropped now, it would land on the target. You may assume no air resistance.

```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main() {

    int height;
    int speed;
    int accel;
    int dist;
    int mass;

    cout << "This program will assume that the aircraft is flying level towards the target. It will also assume no air resistance. All measurements are in standard units." << endl;
    cout << "Please Input Height: " ;
    cin >> height;
    cout << endl;
    cout << "Please Input Speed: " ;
    cin >> speed;
    cout << endl;
    cout << "Please Input Acceleration: " ;
    cin >> accel;
    cout << endl;
    cout << "Please Input Distance from Target: " ;
    cin >> dist;
    cout << endl;

    // calculate time taken for vertical fall if released now
    int time_vert = sqrt ((2 * (height)) / accel);

    if (time_vert <= 0)
    {
        cout << "wrong values entered, ERROR!";
        return 1;
    }

    // calculate horizontal distance travelled if released now before hitting ground
    int hor_dist = (speed * time_vert) + (0.5 * accel * (pow (time_vert, 2)));

    // calculate the acceleration required
    int accel_req = (((dist - (speed * time_vert)) * 2) / pow (time_vert, 2));

    // calculate difference in acceleration
    int accel_diff = accel_req - accel;

    if (accel_diff > 0)
    {
        // acceleration is required
        cout << "Increase Acceleration to " << accel_req << " m/s^2 and drop payload.";
    }

    else if (accel_diff < 0)
    {
        // deceleration is required
        cout << "Decrease Acceleration to " << accel_req << " m/s^2 and drop payload.";
    }

    else
    {
        cout << "Release Now!";
    }


    return 0;
}
```

```
/tmp/4gsNqfjpQT.o
This program will assume that the aircraft is flying level towards the target. It will also assume no air resistance. All measurements are in standard units.
Please Input Height: 100
Please Input Speed: 100
Please Input Acceleration: 3
Please Input Distance from Target: 1000
Increase Acceleration to 6 m/s^2 and drop payload.
```