Rajshahi University of Engineering & Technology

CSE 2102: Sessional Based on CSE 2101

Lab Report 03

Dated: 04.02.18

Submitted to

Rizoan Toufiq

Assistant Professor

Dept. of Computer Science & Engineering

&

Instructor, CSE 2102

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of Computer Science & Engineering

Experiment No. 1

Name of the Experiment: The Foundations: Logic and Proof

1. EXPERIMENT [9]

The 3x+1 Conjecture: Let T be the transformation that sends an even integer x to x/2 and an odd integer x to 3x + 1. A famous conjecture, sometimes known as the 3x + 1 conjecture, states that for all positive integers x, when we repeatedly apply the transformation T, we will eventually reach the integer 1. For example, starting with x = 13, we find $T(13)=3 \cdot 13 + 1 = 40$, $T(40) = 40/2 = 20$, $T(20) = 20/2 = 10$, $T(10) = 10/2 = 5$, $T(5) = 3 \cdot 5 + 1 = 16$, $T(16) = 8$, $T(8) = 4$, $T(4) = 2$,and $T(2) = 1$.

SOLUTION:

```
#include <iostream>

using namespace std;

void conjecture(unsigned long long value)
{
    if(value % 2 != 0){
        value = 3 * value + 1;
        cout << value << endl;
        if(value != 1)
        {
            conjecture(value);
        }
    }
    else{
        value /= 2;
        cout << value << endl;
        if(value != 1)
        {
            conjecture(value);
        }
    }
}

int main()
{
    unsigned long long value;

    cout << "Enter a value: ";
    cin >> value;
```

```
    conjecture(value);
}
```

OUTPUT:

```
Enter a value: 35
106
53
160
80
40
20
10
5
16
8
4
2
1
```

OUTPUT:

```
Enter a value: 10
5
16
8
4
2
1
```

Discussion: The 3x + 1 conjecture is also true for the values greater than 5.6*10^13 as we used unsigned long long variable type here. The output is such **extensive to attach with the report.**

Experiment No. 02

Name of the Experiment: Basic Structure: Sets, Functions, Sequences and Sum

2. EXPERIMENT [2]

Given two finite sets, list all elements in the Cartesian product of these two sets.

SOLUTION:

```
#include <iostream>

using namespace std;

int main()
{
    int ax[] = {1, 2, 3};
    int bx[] = {10, 20, 30};

    cout << "Cartesian Product of the two given sets:\n\n";

    for(int i = 0; i < (sizeof(ax)/4); i++)
    {
        for(int j = 0; j < (sizeof(bx)/4); j++)
        {
            cout << "{" << ax[i] << ", " << bx[j] << "}, ";
        }
    if(i < (sizeof(ax)/4) - 1)
        cout << endl;
    }
}
```

OUTPUT:

```
Cartesian Product of the two given sets:

{1, 10}, {1, 20}, {1, 30},
{2, 10}, {2, 20}, {2, 30},
{3, 10}, {3, 20}, {3, 30},
```

Discussion: All the courtesian product is printed out in pair. Here, as we used nested for loop, the program has the n^2 complexity.

3. EXPERIMENT [6]

Check floor(2x) = floor(x) + floor(x + ½) is true for integer number x = [-100 100].

SOLUTION:

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

using namespace std;

int main()
{
    int flag = 1;
    for(int i = -100; i <= 100; i++)
    {
        if(2*i != (i + floor(i + .5)))
        {
            flag = 0;
        }
    }
    if(flag == 1)
        cout << "The equation is TRUE for all integers from -100
to 100." << endl;
    else
        cout << "The equation is FALSE for all integers from -
100 to 100." << endl;
}
```

OUTPUT:

```
The equation is TRUE for all integers from -100 to 100.
```

Discussion: For all the values from -100 to +100, the equation is true. Again, the equation is also true for wider range as -10000 to 10000 and others.