Rajshahi University of Engineering & Technology

CSE 2102: Sessional Based on CSE 2101

Lab Report 01

Dated: 17.02.18

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

Rajshahi University of Engineering & Technology

CSE 2102: Sessional Based on CSE 2101

Lab Report 02

Dated: 25.02.18

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. 02

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

1. EXPERIMENT [1]

Given subsets A and B of a set with n elements, use bit strings to find A ∪ B, A ∩ B, A – B and A ⊕ B.

THEORY: It is to determine the union, intersection, difference and exclusive-or reguarding to the definition in the topic, Set. Two set was defined as two arrays and later some logical calculations were done reguarding to their function.

```
#include <iostream>

using namespace std;

int main()
{
    int a[] = {1, 2, 3, 4, 5};
    int b[] = {2, 4, 6, 8, 10};

    cout << "A: \t";
    for(int i = 0; i < (sizeof(a)/4); i++)
        cout << a[i] << " ";
        cout << endl << "B: \t";

        for(int i = 0; i < (sizeof(b)/4); i++)
        cout << b[i] << " ";
        cout << endl;

    cout << "\nA U B: \t";

    for(int i = 0; i < (sizeof(a)/4); i++)
    {
        cout << a[i] << ends;
    }

    for(int i = 0; i < (sizeof(a)/4); i++)
    {
        int flag = 0;
        for(int j = 0; j < (sizeof(b)/4); j++)
        {
            if(b[i] == a[j])
            {
                flag = 1;
                continue;
            }
        }
        if(flag == 0)
            cout << b[i] << ends;
```

```
    }
    cout << "\nA : B: \t";

    for(int i = 0; i < (sizeof(a)/4); i++)
    {
        int flag = 0;
        for(int j = 0; j < (sizeof(b)/4); j++)
        {
            if(b[i] == a[j])
            {
                flag = 1;
                cout << b[i] << ends;
            }
        }
    }

    cout << "\nA - B: \t";

    for(int i = 0; i < (sizeof(a)/4); i++)
    {
        int flag = 0;
        for(int j = 0; j < (sizeof(b)/4); j++)
        {
            if(a[i] == b[j])
            {
                flag = 1;
                continue;
            }
        }
        if(flag == 0)
        {
            cout << a[i] << ends;
        }
    }

    cout << "\nA O B: \t";

    for(int i = 0; i < (sizeof(a)/4); i++)
    {
        int flag = 0;
        for(int j = 0; j < (sizeof(b)/4); j++)
        {
            if(a[i] == b[j])
            {
                flag = 1;
                continue;
            }
        }
        if(flag == 0)
        {
            cout << a[i] << ends;
        }
    }
    for(int i = 0; i < (sizeof(b)/4); i++)
    {
        int flag = 0;
        for(int j = 0; j < (sizeof(a)/4); j++)
```

```
        {
            if(b[i] == a[j])
            {
                flag = 1;
                continue;
            }
        }
        if(flag == 0)
        {
            cout << b[i] << ends;
        }
    }
    cout << "\n\nHere, the Union, Intersection, Difference and Exculsive OR
is represented by the symbols U, :, - and O respectively.\n";
}
```

OUTPUT:

```
A:      1 2 3 4 5
B:      2 4 6 8 10

A U B:  1 2 3 4 5 6 8 10
A : B:  2 4
A - B:  1 3 5
A O B:  1 3 5 6 8 10

Here, the union, intersection, difference and Exculsive OR is
represented by the symbols U, :, - and O respectively.
```

2. EXPERIMENT [7]

Find the following summation:

(i) $\sum_{n=L}^{U} (a + n * d)$ **Where L < U, given L, U, a and d.**

THEORY: A for loop was revolving with the variable n from lower limit to upper limit as to determine the summation.

```
#include <iostream>
using namespace std;
int main()
{
    int l, u, a, d, sum = 0;
    cout << "Lower limit: ";
    cin >> l;
    cout << "Upper Limit: ";
    cin >> u;
```

```
    cout << "a: ";
    cin >> a;
    cout << "d: ";
    cin >> d;

    for(int i = l; i <= u; i++)
    {
        sum += a + i*d;
    }

    cout << "\nSummation: " << sum << endl;
    return 0;
}
```

OUTPUT:

```
Lower limit: 2
Upper Limit: 4
a: 1
d: 3

Summation: 30
```

(ii) $\sum_{j=L}^{U} ar^{j}$ Where L < U, given L, U, a and d.

THEORY: A for loop was revolving with the variable j from lower limit to upper limit as to determine the summation.

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

using namespace std;

int main()
{
    int l, u, a, r, sum = 0;
    cout << "Lower limit: ";
    cin >> l;
    cout << "Upper Limit: ";
    cin >> u;
    cout << "a: ";
    cin >> a;
    cout << "d: ";
    cin >> r;

    for(int i = l; i <= u; i++)
```

```
    {
        sum += a*pow(r,i);
    }

    cout << "\nSummation: " << sum << endl;
    return 0;
}
```

OUTPUT:

```
Lower limit: 2
Upper Limit: 4
a: 1
d: 3

Summation: 117
```

(iii) $\sum_{i=L}^{U} \sum_{j=L}^{U} (i+j)$ Where L < U, given L, U.

THEORY: Two for loops was revolving with the variables i and j from lower limit to upper limit as to determine the summation.

```
#include <iostream>

using namespace std;

int main()
{
    int l, u, sum = 0;
    cout << "Lower limit: ";
    cin >> l;
    cout << "Upper Limit: ";
    cin >> u;
    for(int i = l; i <= u; i++)
    {
        for(int j = l; j <= u; j++)
        {
            sum += i + j;
            //cout << i << " + " << j << " = " << sum << endl;
        }
    }

    cout << "\nSummation: " << sum << endl;
    return 0;
}
```

OUTPUT:

```
Lower limit: 2
Upper Limit: 4

Summation: 54
```

3. EXPERIMENT [8]

Find the value of the following series: 1 + 1/2 + 1/4 + 1/8 + 1/16 + ...

THEORY: For the infinite geometric series, the summation is equal to a/(1 - r) where a is the first term, r is the common ratio. This euation is used here to calculate the summation of the series.

For this series, first term a = 1, common ratio = ½ = .5

```
#include <iostream>

using namespace std;

int main()
{
    float sum = 1/(1 - .5);

    cout << "Summation: " << sum << endl;
}
```

OUTPUT:

```
Summation: 2
```

4. EXPERIMENT [9]

Calculate the value of Pi (π)

THEORY: The value of Pi is calculated using three procedures here.

(a) The Gregory-Leibniz Series

(b) Machin's Formula

(c) Archimedes Formula

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

using namespace std;

int main()
{
    /// (a) gregory leibniz series
    int l;
    cout << "Enter n for Gregory Leibniz Series: ";
    cin >> l;

    double pi;

    for(int i = 0; i < l; i++)
    {
        pi += (4*pow(-1, i))/(2*i+1);
    }
    cout << fixed << setprecision(10) << "\nFrom Gregory Leibniz
Series: "  << pi << endl;

    /// (b) Machin's formula
    pi = 16 * atan(1/5.0) - 4*atan(1/239.0);
    cout << fixed << setprecision(10) << "\nFrom Machin's
Formula: "  << pi << "\n\n";


    /// (c) Archimedes Formula
    long n = 6;
    double s = 1.0, a, b, s1, n1;
    while(n < 786423)
    {
        pi = (n*s)/2.0;
        a = sqrt(1 - s*s/4);
        b = 1 - a;
        s1 = sqrt((s * s)/4 + b * b);
        s = s1;
        n1 = 2*n;
        n *= 2;

        cout << fixed << setprecision(10) << s << " \t" << n <<
" \t" << pi << " \t" << a << " \t" << b << " \t" << s1 << " \t"
<< n1 << endl;
    }
```

```
    cout << fixed << setprecision(10) << "\nFrom Archimedes's 
Formula: "  << pi << endl;
}
```

OUTPUT:

```
Enter n for Gregory Leibniz Series: 100

From Gregory Leibniz Series: 3.1315929036

From Machin's Formula: 3.1415926536

0.5176380902    12      3.0000000000    0.8660254038    0.1339745962    0.5176380902    12.0000000000
0.2610523844    24      3.1058285412    0.9659258263    0.0340741737    0.2610523844    24.0000000000
0.1308062585    48      3.1326286133    0.9914448614    0.0085551386    0.1308062585    48.0000000000
0.0654381656    96      3.1393502030    0.9978589232    0.0021410768    0.0654381656    96.0000000000
0.0327234633    192     3.1410319509    0.9994645875    0.0005354125    0.0327234633    192.0000000000
0.0163622792    384     3.1414524723    0.9998661379    0.0001338621    0.0163622792    384.0000000000
0.0081812081    768     3.1415576079    0.9999665339    0.0000334661    0.0081812081    768.0000000000
0.0040906126    1536    3.1415838921    0.9999916334    0.0000083666    0.0040906126    1536.0000000000
0.0020453074    3072    3.1415904632    0.9999979084    0.0000020916    0.0020453074    3072.0000000000
0.0010226538    6144    3.1415921060    0.9999994771    0.0000005229    0.0010226538    6144.0000000000
0.0005113269    12288   3.1415925167    0.9999998693    0.0000001307    0.0005113269    12288.0000000000
0.0002556635    24576   3.1415926194    0.9999999673    0.0000000327    0.0002556635    24576.0000000000
0.0001278317    49152   3.1415926450    0.9999999918    0.0000000082    0.0001278317    49152.0000000000
0.0000639159    98304   3.1415926515    0.9999999980    0.0000000020    0.0000639159    98304.0000000000
0.0000319579    196608  3.1415926531    0.9999999995    0.0000000005    0.0000319579    196608.0000000000
0.0000159790    393216  3.1415926535    0.9999999999    0.0000000001    0.0000159790    393216.0000000000
0.0000079895    786432  3.1415926536    1.0000000000    0.0000000000    0.0000079895    786432.0000000000

From Archimedes's Formula: 3.1415926536
```

5. EXPERIMENT [10]

Calculate the value of Golden ratio.

THEORY: Golden ratio appears frequently in our universe. A simplified equation was used to calculate the value of golden ratio.

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

using namespace std;

int main()
{
    float GR = (1 + sqrt(5)) / 2;

    cout << "Golden Ratio: " << fixed << setprecision(10) << GR 
<< endl;
}
```

OUTPUT:

```
Golden Ratio: 1.6180340052
```