International Islamic University Chittagong (IIUC)

Department of Computer Science Engineering (CSE)

LAB - 2

Course title : **Numerical Methods Lab**

Course code : **CSE-4746**

Session : **Spring-2024**

Submitted To:

Mohammed Shamsul Alam

Professor, Dept. of CSE

International Islamic University Chittagong

Submitted By:

Name : Afif Hossain Irfan

Matric ID : C211005

Semester : 7th

Section : 7AM

Mobile no. : 01521536082

Date of Submission : March 11, 2024

Difference Table.

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ///Peace be with you.


    vector<int> x = {1, 2, 3, 4, 5};
    vector<int> y = {1, 8, 27, 64, 125};
    int n = y.size();


    vector<vector<int>> table(n, vector<int>(n));


    for (int i = 0; i < n; ++i)
    {
        table[i][0] = y[i];
    }

    // 00
    //    01
    // 10    02
    //    11    03
    // 20    12    04
    //    21    13
    // 30    22
    //    31
    // 40

    for (int i = 1; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            table[j][i] = table[j + 1][i - 1] - table[j][i - 1];
        }
    }


    cout << "Difference Table:" << endl;
    cout << "0Y0" << "\t";
    for (int i = 1; i < n; i++)
    {
        cout << i << "Y0" << "\t";
    }
    cout << endl;


    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            cout << table[i][j] << "\t";
        }
        cout << endl;
    }

    return 0;
}
```

LINK : https://www.onlinegdb.com/zA5G2LLuK

Newton's Forward.

```
// Newton Forward

#include <bits/stdc++.h>
using namespace std;

int factorial(int n)
{
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}

int main()
{
    ///Peace be with you.


    vector<double> x = {1, 2, 3, 4, 5};
    vector<double> y = {1, 8, 27, 64, 125};
    double GivenX = 1.7;
    int n = 5;

    vector<vector<double>> table(n, vector<double>(n));

    for (int i = 0; i < n; ++i)
    {
        table[i][0] = y[i];
    }

    for (int i = 1; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            table[j][i] = table[j + 1][i - 1] - table[j][i - 1];
        }
    }

    cout << "Difference Table:" << endl;
    cout << "0Y0" << "\t";
    for (int i = 1; i < n; i++)
    {
        cout << i << "Y0" << "\t";
    }
    cout << endl;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            cout << table[i][j] << "\t";
        }
        cout << endl;
    }

    double u = (GivenX - x[0]) / (x[1] - x[0]); // u = (x-xi)/h

    double ans = y[0];
    for (int i = 1; i < n; i++)
    {
        double backU = u;
        for (int j = 1; j < i; j++)
        {
            backU = backU * (u - j); // backU = u(u-1)(u-2)
        }
```

```
        ans = ans + ((backU * table[0][i]) / factorial(i));
    }

    cout << endl;
    cout << fixed << setprecision(4);
    cout << "Value of y when x = " << GivenX << " is: " << ans << endl;

    return 0;
}
```

LINK : https://www.onlinegdb.com/edit/Mdupz1JnU

Newton's Backward.

```
// Newton Backward

#include <bits/stdc++.h>
using namespace std;

int factorial(int n)
{
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}

int main()
{
    ///Peace be with you.


    vector<double> x = {1, 2, 3, 4, 5};
    vector<double> y = {1, 8, 27, 64, 125};
    double GivenX = 4.7;
    int n = 5;

    vector<vector<double>> table(n, vector<double>(n));

    for (int i = 0; i < n; ++i)
    {
        table[i][0] = y[i];
    }

    for (int i = 1; i < n; i++)
    {
        for (int j = n-1; j >= i; j--)
        {
            table[j][i] = table[j][i - 1] - table[j - 1][i - 1];
        }
    }

    cout << "Difference Table:" << endl;
    cout << "0Y0" << "\t";
    for (int i = 1; i < n; i++)
    {
        cout << i << "Y0" << "\t";
    }
    cout << endl;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j <= i; j++)
        {
            cout << table[i][j] << "\t";
        }
        cout << endl;
    }


    double u = (GivenX - x[n-1]) / (x[1] - x[0]); // u = (x - x[n-1]) / h

    double ans = y[n-1];
    for (int i = 1; i < n; i++)
    {
        double backU = u;
        for (int j = 1; j < i; j++)
        {
            backU = backU * (u + j); // backU = u(u+1)(u+2)...
        }
```

```
        ans = ans + ((backU * table[n-1][i]) / factorial(i));
    }

    cout << endl;
    cout << fixed << setprecision(4);
    cout << "Value of y when x = " << GivenX << " is: " << ans << endl;

    return 0;
}
```

LINK : https://www.onlinegdb.com/edit/EikuoCDfS

Lagrange's Inverse.

```
// Lagrange's
#include <bits/stdc++.h>
using namespace std;


int main()
{
    ///Peace be with you.


    vector<double> x = {1, 2, 3, 4, 5};
    vector<double> y = {1, 8, 27, 64, 125};
    double GivernY = 85;

    int n = 5;

    double ans = 0.0;
    for (int i = 0; i < n; ++i)
    {
        double backY = 1.0;
        for (int j = 0; j < n; ++j)
        {
            if (i != j)
            {
                backY = backY * (GivernY - y[j]) / (y[i] - y[j]);
            }
        }
        ans = ans + (backY * x[i]);
    }

    cout << "Value of x when y = " << GivernY << " is: " << ans << endl;

    return 0;
}
```

LINK : https://onlinegdb.com/qwupN-LHw

Newton’s divided difference.

```
// Dividend Difference
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ///Peace be with you.


    vector<double> x = {1, 3, 4, 6, 10};
    vector<double> y = {0, 18, 58, 190, 920};
    int n = 5;

    vector<vector<double>> table(n, vector<double>(n));

    for (int i = 0; i < n; ++i)
    {
        table[i][0] = y[i];
    }


    for (int i = 1; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            table[j][i] = (table[j + 1][i - 1] - table[j][i - 1]) / (x[i + j] - x[j]);
        }
    }

    cout << "Difference Table:" << endl;
    cout << "0Y0" << "\t";
    for (int i = 1; i < n; i++)
    {
        cout << i << "Y0" << "\t";
    }
    cout << endl;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            cout << table[i][j] << "\t";
        }
        cout << endl;
    }

    double GivenX = 2.7;
    double ans = y[0];
    for (int i = 1; i < n; i++)
    {
        double ForX = 1; // Initialize ForX
        for (int j = 0; j < i; j++)
        {
            ForX = ForX * (GivenX - x[j]); // ForX = (x-x0)(x-x1)...
        }
        ans = ans + (ForX * table[0][i]);
    }

    cout << "Value of y when x = " << GivenX << " is: " << ans << endl;


    return 0;
}
```

LINK : https://onlinegdb.com/NiPiSB1Dp