আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

# Lab Assignment - 2

**Course Title: Numerical Methods Lab**
**Course Code: CSE - 4746**

| Submitted to: | Submitted by: |
|---|---|
| **Prof. Shamsul Alam**<br>**Professor,**<br>**Dept. of CSE.** | **Arnab Barua**<br>**ID:C201010**<br>**7AM**<br>**Dept. of CSE** |

**Q:**

1. The following values of f (x) are given.

| x | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| y = f(x) | 1 | 8 | 27 | 64 | 125 |

Write a program to find difference table for the above values.

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




int main()
{
    vector<int> x = {1, 2, 3, 4, 5};
    vector<int> y = {1, 8, 27, 64, 125};
    vector<int> del_y[10];
    int total = x.size();

    for (int i = 0; i < y.size(); i++) del_y[0].push_back(y[i]);

    for (int i = 1; i < total; i++) {
        for (int j = 0; j + 1 < del_y[i - 1].size(); j++) {
            int first = del_y[i - 1][j];
            int second = del_y[i - 1][j + 1];
            del_y[i].push_back(second - first);
        }
    }

    for (int i = 0; i < total; i++) {
        for (int k = 0; k < i; k++) cout << " ";
        for (int j = 0 ; j < del_y[i].size(); j++) {
            cout << del_y[i][j] << " ";
        }

        cout << "\n";
    }

}
```

**Q:**

2. The following values of f (x) are given.

| x | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| y = f(x) | 1 | 8 | 27 | 64 | 125 |

Write a program to find the values of y when x = 1.7 by using Newton's forward interpolation formula.

```
// Newton forward


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

double u_fact(double u, double n) {
    double mul = 1;
    for (int i = 0; i < n; i++) {
        mul *= (u - i);
    }
    return mul;
}

double fact(double n) {
    double mul = 1;
    for (int i = 1; i <= n; i++) mul *= i;
    return mul;
}

int main()
{
    vector<double> x = {1, 2, 3, 4, 5};
    vector<double> y = {1, 8, 27, 64, 125};
    vector<double> del_y[10];
    double total = x.size();

    for (int i = 0; i < y.size(); i++)
del_y[0].push_back(y[i]);

    for (int i = 1; i < total; i++) {
```

```
        for (int j = 0; j + 1 < del_y[i - 1].size(); j++)
{
            double first = del_y[i - 1][j];
            double second = del_y[i - 1][j + 1];
            del_y[i].push_back(second - first);
        }
    }

    double x_val = 1.7;
    double x0 = 1;
    double u = (x_val - x0) / 1;
    double ans = 0;
    for (int i = 0 ; i < total; i++) {
        double temp = del_y[i][0];
        double dividend = u_fact(u, i);
        double divisor = fact(i);
        ans += (temp * (dividend / divisor));
    }
    cout << "y = " << ans << "\n";
}
```

**Q:**

3. The following values of f (x) are given.

| x | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| y = f(x) | 1 | 8 | 27 | 64 | 125 |

Write a program to find the values of y when x = 4.7 by using Newton's backward interpolation formula.

```
// Newton Backward

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

double u_fact(double u, double n){
    double mul = 1;
    for (int i = 0; i < n; i++) {
        mul *= (u + i);
```

```
    }
    return mul;
}

double fact(double n) {
    double mul = 1;
    for (int i = 1; i <= n; i++) mul *= i;
    return mul;
}

int main()
{
    vector<double> x = {1, 2, 3, 4, 5};
    vector<double> y = {1, 8, 27, 64, 125};
    vector<double> del_y[10];
    double total = x.size();

    for (int i = 0; i < y.size(); i++)
del_y[0].push_back(y[i]);

    for (int i = 1; i < total; i++) {
        for (int j = 0; j + 1 < del_y[i - 1].size(); j++)
{
            double first = del_y[i - 1][j];
            double second = del_y[i - 1][j + 1];
            del_y[i].push_back(second - first);
        }
    }

    for (int i = 0; i < total; i++) {
        for (int k = 0; k < i; k++) cout << " ";
        for (int j = 0 ; j < del_y[i].size(); j++) {
            cout << del_y[i][j] << " ";
        }

        cout << "\n";
```

```
    }

    double x_val = 4.7;
    double xn = 5;
    double u = (x_val - xn) / 1;
    double ans = 0;
    for (int i = 0 ; i < total; i++) {
        double temp = del_y[i][del_y[i].size() - 1];
        double dividend = u_fact(u, i);
        double divisor = fact(i);
        ans += (temp * (dividend / divisor));
    }
    cout << "y = " << ans << "\n";


}
```

**Q:**

4. The following values of f (x) are given.

| x | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| y = f(x) | 1 | 8 | 27 | 64 | 125 |

Write a program to find the values of x for which f (x) = 85 by using Lagrange's inverse interpolation formula.

```
// Langranges

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

double up(double x_val, vector<double> x, int i) {
     int mul = 1;
     for (int j = 0; j < x.size(); j++) {
          if (j == i)continue;
          mul *= (x_val - x[j]);
     }
     return mul;
```

```
}

double down(vector<double> x, int i) {
     int mul = 1;
     for (int j = 0; j < x.size(); j++) {
          if (i == j) continue;
          mul *= (x[i] - x[j]);
     }
     return mul;
}

int main()
{
     vector<double> x = {1, 2, 3, 4, 5};
     vector<double> y = {1, 8, 27, 64, 125};
     /// as it is inverse lets swap x and y.
     x = {1, 8, 27, 64, 125};
     y = {1, 2, 3, 4, 5};
     vector<double> del_y[10];
     double total = x.size();
     double x_val = 85;
     double ans = 0;
     for (int i = 0; i < y.size(); i++) {
          double dividend = up(x_val, x, i);
          double divisor = down(x, i);
          ans += ((dividend / divisor) * y[i]);
     }
     cout << ans << "\n";


}
```

**Q:**

5. The following values of f (x) are given. Prepare the divided difference table for the following data

| | x | 1 | 3 | 4 | 6 | 10 |
|---|---|---|---|---|---|---|
| y = | f(x) | 0 | 18 | 58 | 190 | 920 |

Write a program to find the values of y when x = 2.7 by using Newton's divided difference formula.

```
/// Newtons divided difference


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

double del(double x_val, int n, vector<double> x) {
     double mul = 1;
     for (int i = 0; i < n; i++) {
          mul *= (x_val - x[i]);
     }
     return mul;
}


int main()
{
     vector<double> x = {1, 3, 4, 6, 10};
     vector<double> y = {0, 18, 58, 190, 920};
     vector<double> del_y[10];
     double total = x.size();

     for (int i = 0; i < y.size(); i++)
del_y[0].push_back(y[i]);

     for (int i = 1, k = 1; i < total; i++, k++) {
          for (int j = 0; j + 1 < del_y[i - 1].size();
j++) {
               double firstY = del_y[i - 1][j];
               double secondY = del_y[i - 1][j + 1];
               double firstX = x[j];
               double secondX = x[j + k];
```

```
                del_y[i].push_back( (secondY - firstY) /
(secondX - firstX) );
            }
        }

        for (int i = 0; i < total; i++) {
            for (int k = 0; k < i; k++) cout << " ";
            for (int j = 0 ; j < del_y[i].size(); j++) {
                cout << del_y[i][j] << " ";
            }

            cout << "\n";
        }

        double x_val = 2.7;
        double ans = 0;
        for (int i = 0 ; i < total; i++) {
            double temp = del_y[i][0];
            double del_x = del(x_val, i, x);
            ans += (temp * del_x);
        }
        cout << "y = " << ans << "\n";


}
```