Rajshahi University of Engineering & Technology

CSE 2104: Sessional Based on CSE 2103

Lab Report 07

Dated: 08.04.18

Submitted to

Shyla Afroge

Assistant Professor

Dept. of Computer Science & Engineering

&

Instructor, CSE 2104

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of Computer Science & Engineering

Problem#01: Determining the numerical differentiation of a particulat number from given tabulated function

```
#include <iostream>

#define size 7
#define dySize size - 1
#define ddySize size - 2
#define d3Size size - 3
#define d4Size size - 4
#define d5Size size - 5
#define d6Size size - 6

using namespace std;

int main() {
    double ax[size], ay[size], dy[dySize], ddy[ddySize],
d3y[d3Size], d4y[d4Size], d5y[d5Size], d6y[d6Size];

   for(int i = 0; i < size; i++) {
        cin >> ax[i] >> ay[i];
    }
    for(int i = 0; i < dySize; i++) {
        dy[i] = ay[i + 1] - ay[i];
    }
    for(int i = 0; i < ddySize; i++) {
        ddy[i] = dy[i + 1] - dy[i];
    }
    for(int i = 0; i < d3Size; i++) {
        d3y[i] = ddy[i + 1] - ddy[i];
    }
    for(int i = 0; i < d4Size; i++) {
        d4y[i] = d3y[i + 1] - d3y[i];
    }
    for(int i = 0; i < d5Size; i++) {
        d5y[i] = d4y[i + 1] - d4y[i];
    }
    for(int i = 0; i < d6Size; i++) {
        d6y[i] = d5y[i + 1] - d5y[i];
    }
    cout << "X\tY\tdY\tddY\td3Y\td4Y\td5Y\td6Y\n";
    for(int i = 0; i < size; i++)
    {
        cout << ax[i] << "\t" << ay[i] << "\t" << dy[i] <<
"\t" << ddy[i] << "\t" << d3y[i] << "\t" << d4y[i] << "\t" <<
d5y[i] << "\t" << d6y[i] << endl;
    }
    double inputX;
    int index;
    cout << "Enter x: ";
```

```
    cin >> inputX;
    for(int i = 0; i < size; i++) {
        if(inputX == ax[i]) {
            index = i;
            break;
        }
    }
    cout << "Index = " << index << endl;
    double h = 1 / (ax[1] - ax[0]);

    double diffY = h * (dy[index] - .5*ddy[index] +
d3y[index]/3.0 - .25*d4y[index] + .2*d5y[index]);
    double doublediffY = h * h * (ddy[index] - d3y[index] +
11/12.0 * d4y[index] - 5/6.0 * d5y[index]);
    cout << "dy/dx = " << diffY << endl;
    cout << "ddy/ddx = " << doublediffY << endl;
}
```

OUTPUT:

```
1.0 2.7183
1.2 3.3201
1.4 4.0552
1.6 4.9530
1.8 6.0496
2.0 7.3891
2.2 9.0250
X       Y       dY      ddY     d3Y     d4Y     d5Y     d6Y
1       2.7183  0.6018  0.1333  0.0294  0.0067  0.0013  0.0001
1.2     3.3201  0.7351  0.1627  0.0361  0.008   0.0014  0.0013
1.4     4.0552  0.8978  0.1988  0.0441  0.0094  0.0067  0.0014
1.6     4.953   1.0966  0.2429  0.0535  0.0294  0.008   0.0067
1.8     6.0496  1.3395  0.2964  0.1333  0.0361  0.0094  0.008
2       7.3891  1.6359  0.6018  0.1627  0.0441  0.0294  0.0094
2.2     9.025   2.7183  0.7351  0.1988  0.0535  0.0361  0.0294
Enter x: 1.2
Index = 1
dy/dx = 3.32032
ddy/ddx = 3.31917
```

Problem#02: Determining the maximum and minimun values of a tabulated function

```
#include <iostream>

#define size 5
#define dySize 4
#define ddySize 3
using namespace std;

int main() {
    double x[size], y[size], dy[size], ddy[size], dddy[size],
ddddy[size], bd[size], bdd[size], bddd[size];
    dy[4] = ddy[4] = ddy[3] = dddy[4] = dddy[3] = dddy[2] = bd[0] =
bdd[0] = bdd[1] = bddd[0] = bddd[1] = bddd[2] = 0;

    for(int i = 0; i < size; i++) {
        cin >> x[i] >> y[i];
    }
    for(int i = 0; i < dySize; i++) {
        dy[i] = y[i + 1] - y[i];
    }
    for(int i = 0; i < ddySize; i++) {
        ddy[i] = dy[i + 1] - dy[i];
    }
    cout << "X\tY\tdY\tddY\n";
    for(int i = 0; i < size; i++) {
        cout << x[i] << "\t" << y[i] << "\t" << dy[i] << "\t" <<
ddy[i] << endl;
    }

    int index = 0;
    double h = x[index + 1] - x[index];
    double p = (-2*dy[index]/ddy[index] + 1) / 2.0;
    cout << "p = " << p << endl;
    double maxX = x[index] + p*h;
    cout << "Max_X = " << maxX << endl;

    double xx = maxX;

    double pn = (double) (xx - x[0]) / (x[1] - x[0]);
    dddy[0] = ddy[0] - ddy[1];
    dddy[1] = ddy[1] - ddy[2];
    ddddy[0] = dddy[0] - dddy[1];
    ddddy[1] = ddddy[2] = ddddy[3] = ddddy[4] = 0;

    cout << "X\tY\tdY\tddY\tdddY\tddddY\n";
    for(int i = 0; i < 5; i++)
    {
        dddy[3] = 0;
        cout << x[i] << "\t" << y[i] << "\t" << dy[i] << "\t" <<
ddy[i] << "\t" << dddy[i] << "\t" << ddddy[i] << endl;
    }
```

```
    double fy = x[0] + pn*dy[0] + pn*(pn-1)*ddy[0]/2 + pn*(pn-
1)*(pn-2)*dddy[0]/6;
    cout << "new P = " << pn << "\nMaximum output for MaxX: " << fy
<< endl;
}
```

OUTPUT:

```
1.2 0.9320
1.3 0.9636
1.4 0.9855
1.5 0.9975
1.6 0.9996
X       Y       dY      ddY
1.2     0.932   0.0316  -0.0097
1.3     0.9636  0.0219  -0.0099
1.4     0.9855  0.012   -0.0099
1.5     0.9975  0.0021  0
1.6     0.9996  0       0
p = 3.75773
Max_X = 1.57577
X       Y       dY      ddY     dddY    ddddY
1.2     0.932   0.0316  -0.0097 0.0002  0.0002
1.3     0.9636  0.0219  -0.0099 0       0
1.4     0.9855  0.012   -0.0099 0       0
1.5     0.9975  0.0021  0       0       0
1.6     0.9996  0       0       0       0
new P = 3.75773
Maximum output for MaxX: 1.0023
```