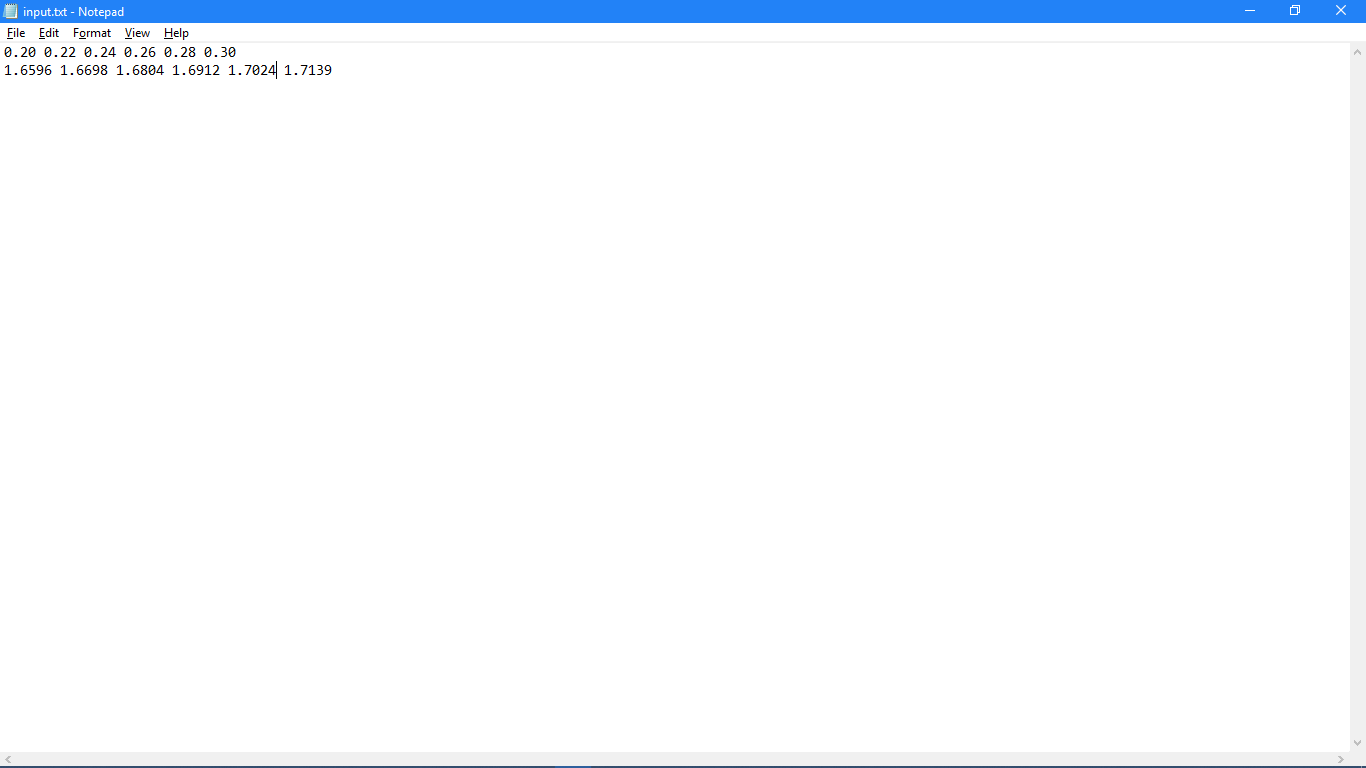
**Problem No: 01**

**Problem Name:** Find f(0.23) and f(0.29) from the following values of x and f(x):



**Objective:** To find functional value from Newton’s Forward and Backward Interpolation Method.

**Source Code:**

#include <iostream>

#include <fstream>

using namespace std;

int n = 6;

double X[6];

double fx[6];

double delx[5], del2x[4], del3x[3], del4x[2], del5x[1];

double mid = 0.24;

void fd(double x);

void bd(double x);

double cf(double p, int i);

double cb(double p, int i);

double fact(int i);

int main()

{

int no;

ifstream fi;

fi.open("input.txt");

for(int i = 0; i < 6; i++)

fi >> X[i];

for(int i = 0; i < 6; i++)

fi >> fx[i];

double x1;

int i;

for(i = 0; i < n; i++)

delx[i] = fx[i + 1] - fx[i];

for(i = 0; i < n - 2; i++)

del2x[i] = delx[i + 1] - delx[i];

for(i = 0; i < n - 3; i++)

del3x[i] = del2x[i + 1] - del2x[i];

for(i = 0; i < n - 4; i++)

del4x[i] = del3x[i + 1] - del3x[i];

for(i = 0; i < n - 5; i++)

del5x[i] = del4x[i + 1] - del4x[i];

cout << "Difference Table:\n";

for(i = 0; i < n; i++){

cout << X[i] << "\t" << fx[i] << "\t";

if(i < n - 1)

cout << delx[i] << "\t";

if(i < n - 2)

cout << del2x[i] << "\t";

if(i < n - 3)

cout << del3x[i] << "\t";

if(i < n - 4)

cout << del4x[i] << "\t";

if(i < n - 5)

cout << del5x[i] << "\t";

cout << endl;

}

while(1){

cout << "\nEnter value: (Enter 0 to exit) ";

cin >> x1;

if(x1 < mid && x1 != 0){

fd(x1);

}

else if(x1 > mid && x1 != 0){

bd(x1);

}

else if(x1 == 0)

break;

}

return 0;

}

void fd(double x)

{

double p, x0, h = 0.02, y;

if(x > 0.20 && x < 0.22)

x0 = 0.20;

else if(x > 0.22 && x < 0.24)

x0 = 0.22;

p = (x - x0) / h;

cout << "\nForward Difference Formula:\n";

cout << "x: " << x << endl;

cout << "x0: " << x0 << endl;

cout << "h: " << h << endl;

cout << "p: " << p << endl;

y = fx[1] + p \* delx[1] + (cf(p, 1) / 2) \* del2x[1] +

(cf(p, 2) / fact(3)) \* del3x[1] + (cf(p, 3) / fact(4)) \* del4x[1];

cout << "Result: " << y << endl << endl;

}

void bd(double x)

{

double p, xn, h = 0.02, y;

if(x > 0.24 && x < 0.26)

xn = 0.26;

else if(x > 0.26 && x < 0.28)

xn = 0.28;

else if(x > 0.28 && x < 0.3)

xn = 0.3;

p = (x - xn) / h;

cout << "\nBackward Difference Formula:\n";

cout << "x: " << x << endl;

cout << "xn: " << xn << endl;

cout << "h: " << h << endl;

cout << "p: " << p << endl;

y = fx[n - 1] + p \* delx[n - 2] + (cb(p, 1) / 2) \* del2x[n - 3] + (cb(p, 2) / fact(3)) \* del3x[n - 4]

+ (cb(p, 3) / fact(4)) \* del4x[n - 5] + (cb(p, 4) / fact(5)) \* del5x[n - 6];

cout << "Result: " << y << endl << endl;

}

double cf(double p, int i)

{

int j;

double p1 = 1;

for(j = 0; j <= i; j++ )

p1 \*= (p - j);

return p1;

}

double cb(double p, int i)

{

int j;

double p1 = 1.0;

for(j = 0; j <= i; j++ )

p1 \*= (p + j);

return p1;

}

double fact(int i)

{

double j, f = 1.0;

for(j = 1.0; j < i; j++)

f \*= j;

return f;

}

**Output:**

