**Name of The Experiment**: Numerical Differentiation.

**Introduction:**The general method for deriving the numerical formulae is to differentiate the interpolating polynomial. We illustrate the derivation with Newton’s forward difference formula has illustrated only, the method of derivation being the same with regard to the other formulae. Newton’s forward difference formula:

y=y0+u∆y0+∆2y0+∆3y0+……………….

Where x=x0+uh.

Then =(∆y0+∆2y0+∆3y0+……) …………(i)

This formula can be used for computing the value of dy/dx for non-tabular values of x. by setting x=x0,

[(∆y0-∆2y0+∆3y0-∆4y0…) ………(ii)

Differentiating eqn(i) once again,

=(∆2y0+∆3y0+∆4y0­+…) ……….(iii)

From which we obtain, when u=0,

[(∆2y0-∆3y0+∆4y0+…) …………(iv)

Formulae for computing higher derivatives may be obtained by successive differentiation. Similarly different formulas can be derived.

**Source Code:**

#include<bits/stdc++.h>

using namespace std;

int main()

{

float x[100],y[100][100],xn,u,h,d1f,d2f,d3f;

int n,i,j;

cout<<"Enter total number of values n = "<<endl;

cin>>n;

cout<<"X Y\n";

for(i=1;i<=n;i++)

{

cin>>x[i]>>y[i][0];

}

for(j=1;j<n;j++){

for(i=1;i<=(n-j);i++)

{y[i][j] = y[i+1][j-1] - y[i][j-1];

}

}

cout<<"\n Forward Difference Table \n";

for(i=1;i<=n;i++)

{

printf("X[%d]=%.4f Y[%d][0]=%.4f \t",i,x[i],i,y[i][0]);

for(j=1;j<=(n-i);j++)

printf(" y[%d][%d]=%.4f",i,j,y[i][j]);

printf("\n");

}

printf("\nEnter x = ");

cin>>xn;

h=x[2]-x[1];

u=(xn-x[1])/h;

cout<<"\nThe value of u = "<<u<<" and h="<<h<<"\n";

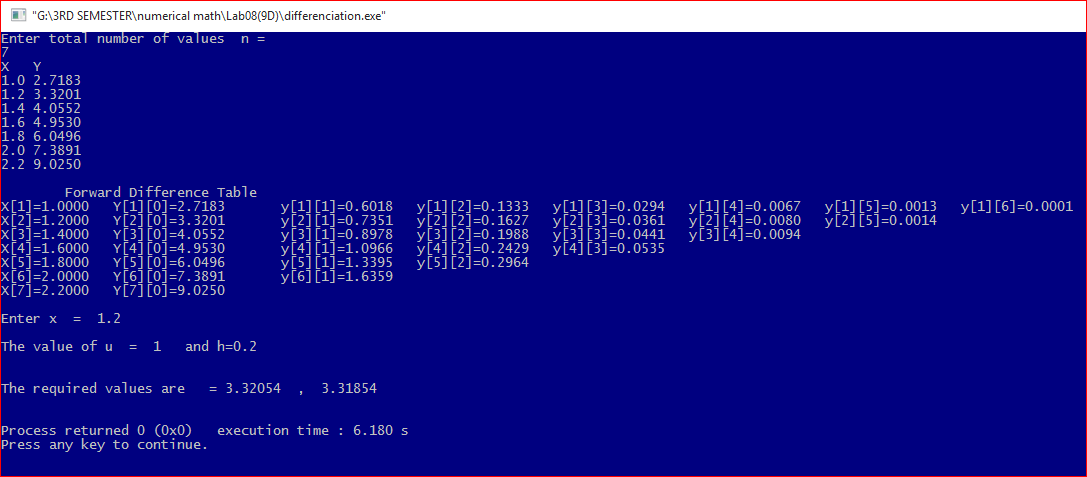
d1f=(1/h)\*(y[1][1]+(2\*u-1)\*y[1][2]/2+(3\*pow(u,2)-6\*u+2)\*y[1][3]/6+(4\*pow(u,3)-18\*pow(u,2)+22\*u-6)\*y[1][4]/24);

d2f=(1/(h\*h))\*(y[1][2]+(u-1)\*y[1][3]+(6\*pow(u,2)-18\*u+11)\*y[1][4]/12);

cout<<endl<<"\nThe required values are = "<<d1f<<" , "<<d2f<<"\n\n";

}

**Input/Output:**

****

**Discussion:**

The general method for deriving the numerical formulae is to differentiate the interpolating polynomial. It is necessary to derive formula to solve the problems with numerical differentiation. The numerical differentiation formula has been shown for first and second order derivatives with Newton’s forward difference formula. Different formula can be derived by starting with other interpolation formulae.