**Interpolation**

**Newton’s Forward And Backward Difference Formula**

**Theory**

If y0 ,y1, y2 , ……. yn denote a set of values of y, then y1 – y0 , y2 – y1 , ….. yn – yn-1 are called the first forward differences of y. Denoting these differences by ∆y0 , ∆y1 ,….. ∆yn-1 respectively we have,

∆y0 = ∆y1 - ∆y0 , ∆y1 = ∆y2 - ∆y1 , ∆y2 = ∆y3 - ∆y2 , ……………….. ∆yn-1 = ∆yn - ∆yn-1

h = x0 – x1 , P =

Where ∆ is called the Forward Difference Operator. And ∆y1 , ∆y0 are called first forward differences. The differences of first forward differences are called second forward differnces and are denoted by ∆2y0 , ∆2y1 …… ∆2yn , Similarly third, fourth, fivth ,…… nth forward differences can be determined. Forward Difference formula for finding the unknown value of y(x) for the value of x is :

yn(x) = y0+P∆y0++ …….. (1)

If y0 ,y1, y2 , ……. yn denote a set of values of y, then y1 – y0 , y2 – y1 , ….. yn – yn-1 are called the first backward differences of y. Denoting these differences by ▼y0 , ▼y1 ,….. ▼yn-1 respectively we have,

▼y0 = ▼y1 - ▼y0 , ▼y1 = ▼y2 - ▼y1 , ▼y2 = ▼y3 - ▼y2 , ………….. ▼yn-1 = ▼yn - ▼yn-1

h = x0 – x1 , P =

Where ▼ is called the backward Difference Operator. And ▼y1 , ▼y0 are called first backward differences. The differences of first backward differences are called second backward differnces and are denoted by ▼2y0 , ▼2y1 …… ▼2yn , Similarly third, fourth, fivth ,…… nth backward differences can be determined. backward Difference formula for finding the unknown value of y(x) for the value of x is :

yn(x) = yn+P▼yn+▼2yn + ▼3yn +…-.+ ▼nyn ….. (2)

**Code**

#include<bits/stdc++.h>

using namespace std;

int fact(int n)

{

if(n==0 || n==1)

return 1;

else if(n>1)

return n\*fact(n-1);

}

void PrintForTable(double x[],double y[],double Del1[], double Del2[], double Del3[])

{

int i,j;

cout<<"x y Del1y Del2y Del3y"<<endl;

cout<<"----------------------------------------------------------"<<endl;

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

{

printf("%0.3lf %0.3lf %0.3lf %0.3lf %0.3lf\n",x[i],y[i],Del1[i],Del2[i],Del3[i]);

}

printf("\n");

}

void PrintBackTable(double x[],double y[],double Del1[], double Del2[], double Del3[], double Del4[], double Del5[])

{

int i,j;

cout<<"x y Del1y Del2y Del3y Del4y Del5y"<<endl;

cout<<"----------------------------------------------------------------------------------------------------------"<<endl;

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

{

printf("%0.7lf %0.7lf %0.7lf %0.7lf %0.7lf %0.7lf %0.7lf\n",x[i],y[i],Del1[i],Del2[i],Del3[i],Del4[i],Del5[i]);

}

printf("\n");

}

void Forward()

{

int points,flag=0,i,j=0,k=0,l=0;

double x[10],y[10],h,p,X,Del1[10],Del2[10],Del3[10],Y;

for(int x=0;x<10;x++)

{

Del1[x] = 0; Del2[x] = 0; Del3[x] = 0;

}

cout<<"\nHow many points? : ";

cin>>points; printf("\n");

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

{

cout<<"Enter the value of x"<<i<<" : " ;

cin>>x[i];

cout<<"Enter the value of y"<<i<<" : ";

cin>>y[i];

}

printf("\n");

i = 0;

while(i<points-1)

{

Del1[i] = y[i+1] - y[i];

i++;

}

i = 0;

while(i<points-2)

{

Del2[i] = Del1[i+1] - Del1[i];

i++;

}

i = 0;

while(i<points-3)

{

Del3[i] = Del2[i+1] - Del2[i];

i++;

}

cout<<"For which value of x, the value of y will be determined: ";

cin>>X;

h = x[1] - x[0];

p = (X - x[0])/h;

printf("\n");

PrintForTable(x,y,Del1,Del2,Del3);

Y = y[0] + (p/fact(1))\*Del1[0] + ((p\*(p-1))/fact(2))\*Del2[0] + ((p\*(p-1)\*(p-2))/fact(3))\*Del3[0];

cout<<"The value of Y("<<X<<") is: " <<Y<<endl;

}

void Backward()

{

int points,flag=0,i,j=0,k=0,l=0;

double x[10],y[10],h,p,X,Del1[10],Del2[10],Del3[10],Del4[10],Del5[10],Y;

for(int x=0;x<10;x++)

{

Del1[x] = 0;

Del2[x] = 0;

Del3[x] = 0;

Del4[x] = 0;

Del5[x] = 0;

}

cout<<"\nHow many points? : ";

cin>>points;

printf("\n");

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

{

cout<<"Enter the value of x"<<i<<" : " ;

cin>>x[i];

cout<<"Enter the value of y"<<i<<" : ";

cin>>y[i];

}

printf("\n");

h = x[1] - x[0];

cout<<"For which value of x, the value of y will be determined: ";

cin>>X;

p = (X - x[points-1])/h;

i = 0;

while(i<points-1)

{

Del1[i] = y[i+1] - y[i];

i++;

}

i = 0;

while(i<points-2)

{

Del2[i] = Del1[i+1] - Del1[i];

i++;

}

i = 0;

while(i<points-3)

{

Del3[i] = Del2[i+1] - Del2[i];

i++;

}

i = 0;

while(i<points-4)

{

Del4[i] = Del3[i+1] - Del3[i];

i++;

}

i = 0;

while(i<points-5)

{

Del5[i] = Del4[i+1] - Del4[i];

i++;

}

printf("\n");

PrintBackTable(x,y,Del1,Del2,Del3,Del4,Del5);

int m = (int) X;

Y = y[--points] + (p/fact(1))\*Del1[--points] + ((p\*(p+1))/fact(2))\*Del2[--points] + ((p\*(p+1)\*(p+2))/fact(3))\*Del3[--points];

printf("The Value of Y(%d) is: %0.7lf\n",m,Y);

}

void menu()

{

int choice;

cout<<"1. Forward Interpolation\n2. Backward Interpolation\n3. Exit\nEnter your Choice: ";

cin>>choice;

switch(choice)

{

case 1:

Forward(); break;

case 2:

Backward(); break;

case 3:

cout<<"The program ended successfully" <<endl; break;

default:

cout<<"wrong Input"<<endl; break;

}

}

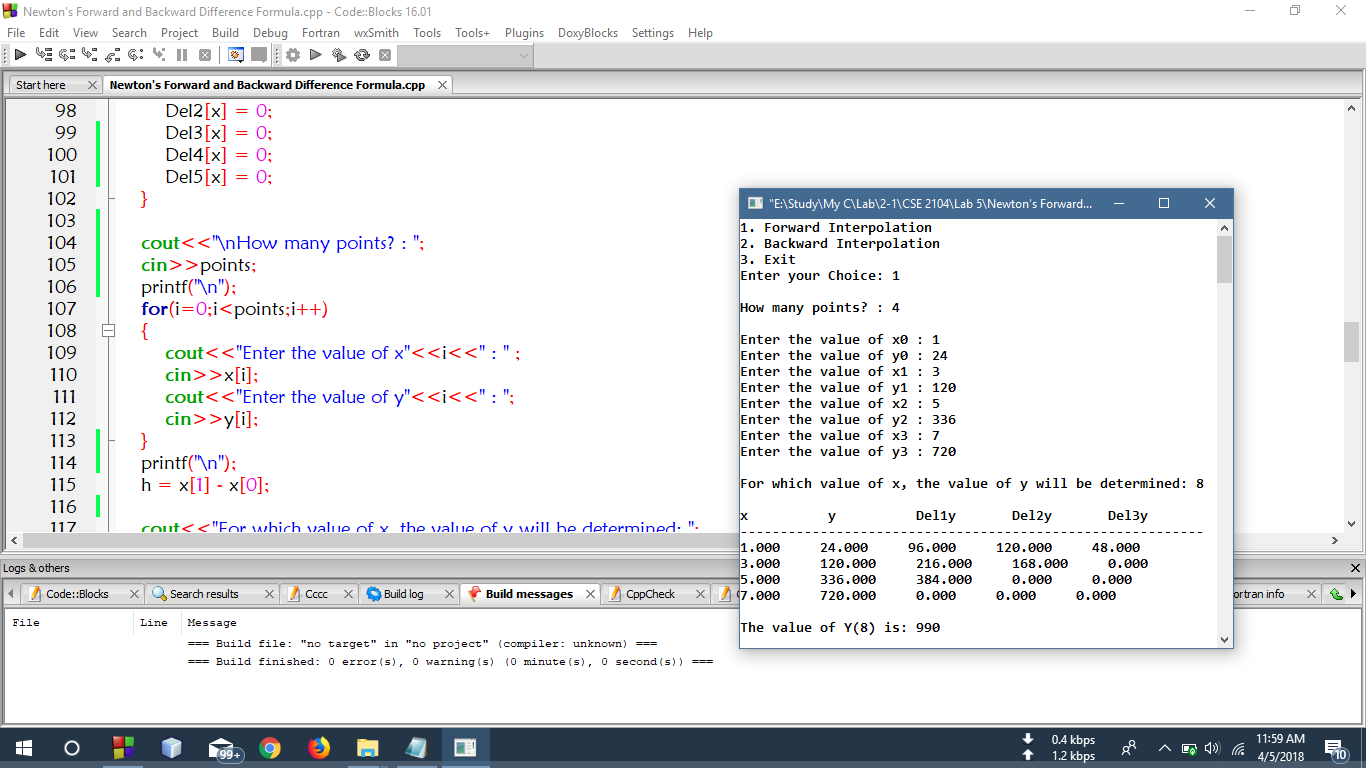
int main()

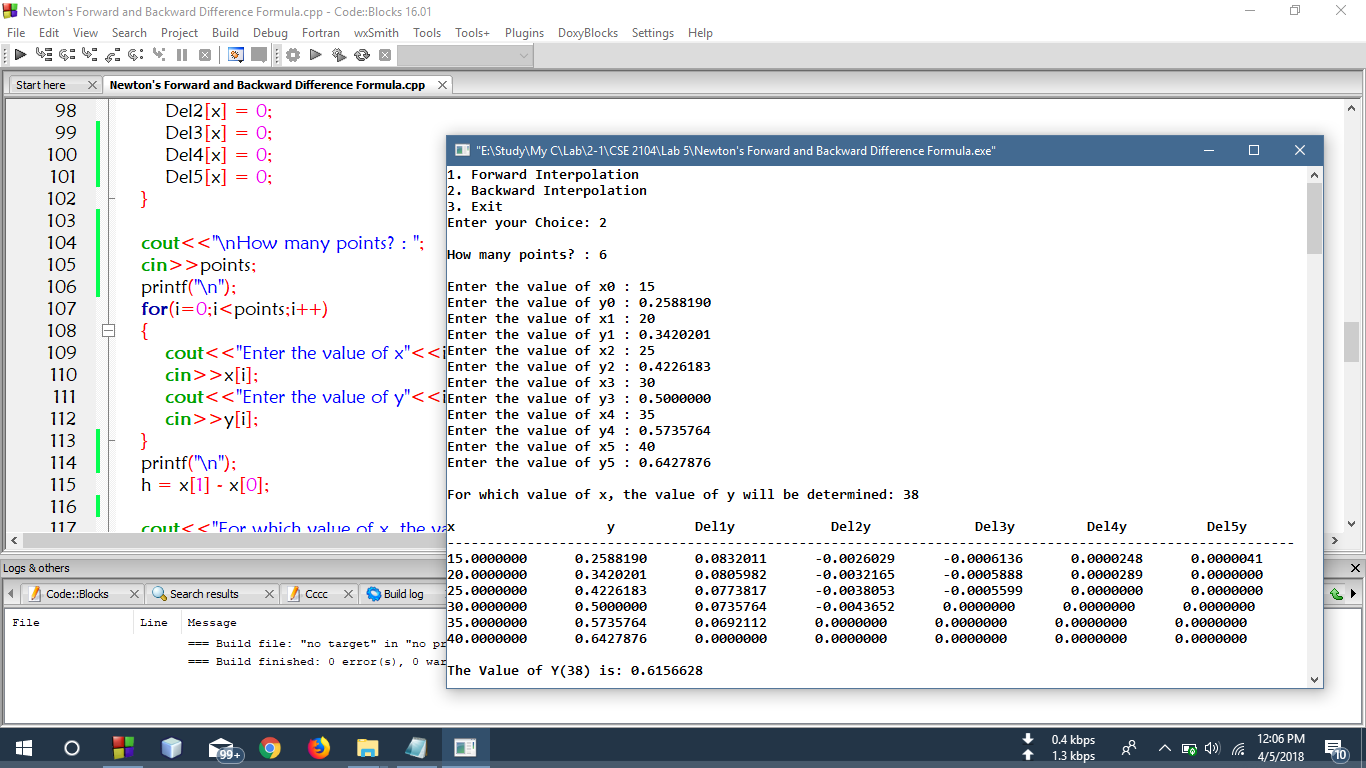
{

menu();

return 0;

}

**Output**



**Discussion**

Here, in the above code , A menu() function was used to get the forward difference table, and backward Difference table which was used to get the required result. In the menu() function, Forward(), Backward() function was called. In the Forward() function, PrintForTable() was called to generate the Forward Difference table and similarly PrintBackTable() was called in the Backward() functionto generate the backward difference table. Thus Using the equation (1) and (2) the result for forward difference and backward difference was shown respectfully.