Name : Deepankar Sharma

Course: BCA

University Roll No: 2092014

Student Id : 20041299

Semester: 3

Date: December 12, 2021

**Practical 12: Newton Backward Interpolation Formula**

**Objective:**To find value of y corresponding given value of x using Newton Backward Interpolation Formula.

**2. Algorithm:**

1. Start
2. Input the matrix of equations in arr[n][n+1] where n is number of given variables in data.

3. Calculate the difference table using:

(j = 2; j < n+1; j++)

{

for (i = 0; i < n-j+1; i++)

{

mat[i][j]= mat[i+1][j-1]- mat[i][j-1];

}

}

4. Enter the value of x

5. Set value of p =x-mat[0][0]/(mat[1][0]-mat[0][0])

6. p0=p

7. Find value of y using:

sum=0.0;

sum=sum+mat[0][1];

k=1;

for (j = 2; j < n+1; j++)

{

sum+= (p\*mat[0][j])/k;

k\*=j;

p\*=(p0-j+1);

}

8. y= sum

9. Print y

10. Stop

**Code:**

#include<iostream>

using namespace std;

int main(){

int n;

printf("Enter the number of known variables: ");

cin>>n;

// float mat[n][n+1]={0.0};

static float mat[10][10]={0.0};

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

{

/\* code \*/

printf("Enter x%d: ", i+1);

cin>>mat[i][0];

printf("Enter y%d: ", i+1);

cin>>mat[i][1];

}

cout<<"x"<<'\t'<<"y"<<endl<<endl;

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

{

/\* code \*/

cout<<mat[i][0]<<'\t'<<mat[i][1]<<endl;

}

// difference table

for (int j = 2; j < n+1; j++)

{

/\* code \*/

for (int i = 0; i < n-j+1; i++)

{

/\* code \*/

mat[i][j]= mat[i+1][j-1]- mat[i][j-1];

}

}

// displaying the difference table

cout<<"\nDisplaying the difference table: "<<endl<<endl;

cout<<"x"<<'\t'<<"y"<<'\t'<<"dy1"<<'\t'<<"dy2"<<'\t'<<"dy3"<<'\t'<<"dy4"<<'\t'<<"dy5";

cout<<endl<<endl;

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

{

/\* code \*/

// if(i<2)

for (int j = 0, j\_dash=n; j\_dash >= 0; j++, j\_dash--)

{

/\* code \*/

cout<<mat[i][j]<<'\t';

}

// else

cout<<endl;

}

// applying the Newton forward formula

float x;

cout<<"Enter the value of x at which you want to calculate the value of y: ";

scanf(" %f", &x) ;

float p= x-mat[0][0];

p/=(mat[1][0]-mat[0][0]);

float p0=p;

float sum=0.0;

sum=sum+mat[0][1];

int k=1;

for (int j = 2; j < n+1; j++)

{

/\* code \*/

sum+= (p\*mat[0][j])/k;

k\*=j;

p\*=(p0-j+1);

}

printf("Value of y at x=%f is: %f", x, sum);

return 0;

}

**Output:**

PS D:\01\_Java\Deepankar\CCpp\CBNST\Practical-no-11-NewtonForwardInterpolation> cd "d:\01\_Java\Deepankar\CCpp\CBNST\Practical-no-11-NewtonForwardInterpolation\" ; if ($?) { g++ 11\_newtonForwardInterpolation.cpp -o 11\_newtonForwardInterpolation } ; if ($?) { .\11\_newtonForwardInterpolation }

Enter the number of known variables: 5

Enter x1: 2

Enter y1: 4

Enter x2: 4

Enter y2: 8

Enter x3: 6

Enter y3: 12

Enter x4: 8

Enter y4: 16

Enter x5: 10

Enter y5: 20

x y

2 4

4 8

6 12

8 16

10 20

Displaying the difference table:

x y dy1 dy2 dy3 dy4 dy5

2 4 4 0 0 0

4 8 4 0 0 0

6 12 4 0 0 0

8 16 4 0 0 0

10 20 0 0 0 0

Enter the value of x at which you want to calculate the value of y: 5

Value of y at x=5.000000 is: 10.000000