**ASSIGNMENT NO.**

**PROBLEM STATEMENT:**

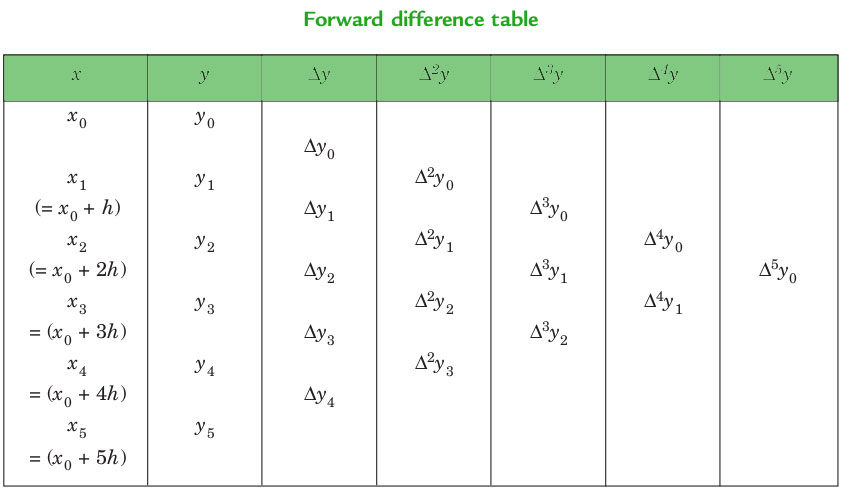
Program in C to find out value of f(x) using Newton’s Forward Interpolation.

**THEORY:**

**Interpolation** is the technique of estimating the value of a function for any intermediate value of the independent variable, while the process of computing the value of the function outside the given range is called **extrapolation**.

**Forward Differences** : The differences y1 – y0, y2 – y1, y3 – y2, ……, yn – yn–1 when denoted by dy0, dy1, dy2, ……, dyn–1 are respectively, called the first forward differences. Thus the first forward difference are:





**Newton’s\_Forward\_Interpolation\_Formula**:



This formula is particularly useful for interpolating the values of f(x) near the beginning of the set of values given. h is called the interval of difference and **u = ( x – a ) / h**, Here a is first term.

**ALGORITHM:**

**Input:** 1) The total number of records user wants to enter(say n), the values of predefined variables, say x0,x1,.....xn-1 and their respective functional values.

2) An intermediate value of x for which the functional value is to be estimated.

**Output:** Estimated functional value of x.

**Data Structure Used:** Two arrays to store different variables and their respective functional values.

**Steps: Algorithm for main() function**

STEP 1: set k = 0

STEP 2: set p = xi // the intermediate value of x for which functional value is to be //approximated.

STEP 3: set i=1

STEP 4: While(i<n)

Begin

STEP 5: set j=0

STEP 6: While(j<n-i)

Begin

STEP 7: set y[i][j] = y[i-1][j+1] – y[i-1][j]

STEP 8: set j= j+1

End While

STEP 9: set i=i+1

End While

STEP 10: set i=0

STEP 11: While(i<n)

Begin

STEP 12: PRINT “ x ” //every value of x for which f(x) is provided.

STEP 13: Set j=0;

STEP 14: While(j<n-i)

Begin

STEP 15: PRINT “ f(x) ” //for each x

STEP 16: set j=j+1

End While

STEP 17: set i=i+1

End While

STEP 18: set i=0

STEP 19: Repeat from Step 20 to step 22 while(k!=1)

STEP 20: If(x[i]<p && p<x[i+1])

Then

STEP 21: set k = 1

Else

STEP 22: set i=i+1

End If

STEP 23: set u = (p-x[f])/(x[f+1]-x[f])

STEP 24: set n = n-i+1

STEP 25: set sum = 0

STEP 26: set i = 0

STEP 27: While(i<n-1)

Begin

STEP 28: set temp = 1

STEP 29: set j = 0

STEP 30: While(j<i)

Begin

STEP 31: set temp temp \* (u-j)

STEP 32: set j = j+1

End While

STEP 33: set m = fact(i)

STEP 34: set sum = sum + temp \* (y[i][f] / m)

STEP 35: set i = i+1

End While

**Algorithm for fact(a) function** // a is any integer which act as an argument to this function

STEP 1: set float fac = 1

STEP 2: If ( a == 0 )

Then

STEP 3: return (1)

Else

STEP 4: fac = a \* fact(a-1)

End If

STEP 5: return (fac)

**SOURCE CODE:**

#include<stdio.h>

int main()

{

float x[10],y[10][10],sum,p,u,temp;

int i,n,j,k=0,f,m;

float fact(int);

printf("\nHow many records you want to enter:");

scanf("%d",&n);

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

{

printf("\nEnter the value of x%d: ",i);

scanf("%f",&x[i]);

printf("\nEnter the value of f(x%d):",i);

scanf("%f",&y[k][i]);

}

printf("\nEnter X for finding f(x):");

scanf("%f",&p);

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

{

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

{

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

}

}

printf("\nThe Table is:\n\n");

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

{

printf("\n %.3f ",x[i]);

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

{

printf(" ");

printf(" %.3f ",y[j][i]);

}

printf("\n");

}

i=0;

do{

if(x[i]<p && p<x[i+1])

k=1;

else

i++;

}while(k!=1);

f=i;

u=(p-x[f])/(x[f+1]-x[f]);

printf("\n\n u= %.3f",u);

n-n-i+1;

sum=0;

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

{

temp=1;

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

{

temp= temp \* (u-j);

}

m=fact(i);

sum= sum +temp \* (y[i][f]/m);

}

printf("\n\n f(%.2f) = %f",p,sum);

}

float fact(int a)

{

float fac = 1;

if(a==0)

return 1;

else

fac = a \* fact(a-1);

return (fac);

}

**INPUT AND OUTPUT :**

How many records you want to enter:5

Enter the value of x0: 2

Enter the value of f(x0):9

Enter the value of x1: 2.25

Enter the value of f(x1):10.06

Enter the value of x2: 2.5

Enter the value of f(x2):11.25

Enter the value of x3: 2.75

Enter the value of f(x3):12.56

Enter the value of x4: 3

Enter the value of f(x4):14

Enter X for finding f(x):2.35

The Table is:

2.000 9.000 1.060 0.130 -0.010 0.020

2.250 10.060 1.190 0.120 0.010

2.500 11.250 1.310 0.130

2.750 12.560 1.440

3.000 14.000

u= 0.400

f(2.35) = 10.522240

**DISCUSSION :**

**1)** We can see in our program we used array whose size is to be predefined. It will be convenient if some dynamic memory would be alloted for our required values of x and f(x) to be stored in. This will hault the problem of wastage of memory/scarcity of memory if array size is predefined.