Newton's Divided Difference formula was put forward to overcome a few limitations of Lagrange's formula. In Lagrange's formula, if another interpolation value were to be inserted, then the interpolation coefficients were to be calculated again. This is not the case in Divided Difference. In this tutorial, we're going to discuss a source code in C for Newton Divided Difference formula along with sample output.

Newton's Divided Difference Formula eliminates the drawback of recalculation and recomputation of interpolation coefficients by using Newton's general interpolation formula which uses "divided differences". Before going through the source code for **Newton Divided Difference in C**, here's a brief explanation of what divided differences are with the formula for divided differences.

If (x0,y0), (x1,y1), (x2,y2) …… be given points, then the first divided difference for the arguments x0, x1 is defined as: [x0, x1] = (y1-y0)/(x1-x0). And, similarly, [x1, x2] = (y2-y1)/(x2-x1), and after that [x2, x3] = (y3-y2)/(x3-x2), and so on.

$$f(x) = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + \ldots + (x - x_0)(x - x_1) \ldots (x - x_{k-1}) f[x_0, x_1, \ldots, x_k].$$

Newton's Divided Difference Formula

The second divided difference is defined as: [x0, x1, x2] = ([x1, x2] – [x0, x1])/(x2-x0). This goes on in similar fashion for the third, fourth …. and nth divided differences. Based on these formulas, two basic properties of Newton's Divided Difference method can be outlined as given below:

- The divided differences are symmetrical in their arguments i.e. independent of the order of the arguments.
- The nth divided differences of a polynomial of the nth degree are constant.

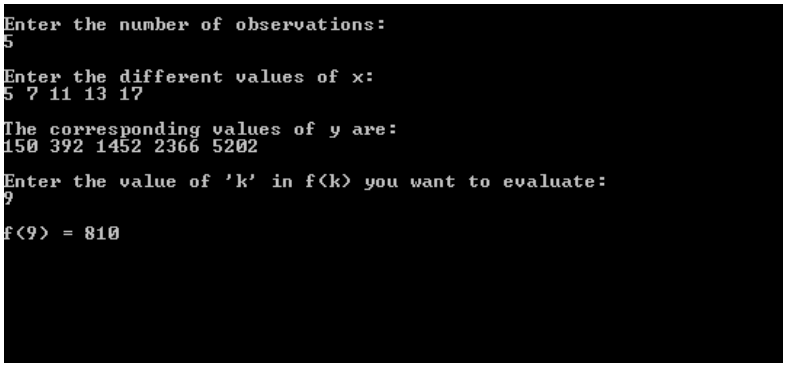
Newton Divided Difference in C:

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int x[10], y[10], p[10];
    int k,f,n,i,j=1,f1=1,f2=0;
    printf("\nEnter the number of observations:\n");
    scanf("%d", &n);

    printf("\nEnter the different values of x:\n");
    for (i=1;i<=n;i++)
        scanf("%d", &x[i]);
```

```
    do
    {
        for (i=1;i<=n-1;i++)
        {
            p[i] = ((y[i+1]-y[i])/(x[i+j]-x[i]));
            y[i]=p[i];
        }
        f1=1;
        for(i=1;i<=j;i++)
            {
                f1*=(k-x[i]);
            }
        f2+=(y[1]*f1);
        n--;
        j++;
    }

    while(n!=1);
    f+=f2;
    printf("\nf(%d) = %d", k , f);
    getch();
}
```

This C program first asks for the number of observations. It then asks for different values of x and the corresponding values of y. Then, you need to enter the value of "k" in f(k) you want to evaluate. The Newton Divided Difference calculation is done using do-while loop.

Input/Output:



Also see,

C Program for Lagrange Interpolation

C Program for Newton Forward Interpolation