NEWTONS DIVIDED DIFFERENCE METHOD

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
  float x[10],y[10][10],sum,p,u,temp;
  int i,n,j,k=0,f,m;
  float fact(int);
  clrscr();
  printf(\"\\nhow many record you will be enter: \");
  scanf(\"%d\",&n);
  for(i=0; i<n; i++)
  {
   printf(\"\\n\\nenter the value of x%d: \",i);
   scanf(\"%f\",&x[i]);
   printf(\"\\n\\nenter the value of f(x%d): \",i);
   scanf(\"%f\",&y[k][i]);
  }
  printf(\"\\n\\nEnter X for finding f(x): \");
  scanf(\"%f\",&p);

  for(i=1;i<n;i++)
  {
    k=i;
    for(j=0;j<n-i;j++)
    {
     y[i][j]=(y[i-1][j+1]-y[i-1][j])/(x[k]-x[j]);
     k++;
    }
  }
  printf(\"\\n_____________________________________________________\\n\");
  printf(\"\\n  x(i)\\t   y(i)\\t    y1(i)    y2(i)    y3(i)    y4(i)\");
  printf(\"\\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;

  sum=0;
  for(i=0;i<n-1;i++)
  {
   k=f;
   temp=1;
```

```
   for(j=0;j<i;j++)
   {
    temp = temp * (p - x[k]);
    k++;
   }
    sum = sum + temp*(y[i][f]);
  }
  printf(\"\\n\\n f(%.2f) = %f \",p,sum);
  getch();
}


/*
_____________________________________________

         OUT PUT
_____________________________________________


how many record you will be enter: 5



enter the value of x0: 2.5


enter the value of f(x0): 8.85


enter the value of x1: 3


enter the value of f(x1): 11.45


enter the value of x2: 4.5


enter the value of f(x2): 20.66


enter the value of x3: 4.75


enter the value of f(x3): 22.85


enter the value of x4: 6


enter the value of f(x4): 38.60


Enter X for finding f(x): 3.5

_____________________________________________________________

  x(i)     y(i)     y1(i)    y2(i)    y3(i)    y4(i)
_____________________________________________________________
```

```
 2.500    8.850     5.200    0.470    0.457    -0.029

 3.000    11.450    6.140    1.497    0.354

 4.500    20.660    8.760    2.560

 4.750    22.850    12.600

 6.000    38.600


 f(3.50) = 13.992855
*/
```