ASSIGNMENT NO:7 **DATE:27/02/2015**

PROGRAM STATEMENT:Compute the positive root of x^3-x-.1=0 by Newton-Raphson Method,correct upto 6 significant figures.

THEORY: This is an iterative method. Let **$x=(a_0<=x_0<=b_0$**) is an approximation of the root (alpha) of the equation f(x)=0.Let h be a small correction on x_0 ,then $x_1=x_0+h$ is the correct root.

So,$f(x_1)=0=>f(x_0+h)=0$.

Therefore ,by Taylor's Theorem,we get,

$f(x_0)+hf'(x_0)+\cdots\cdots=0$.

As h is small neglating the second and higher order power of h,we get,

$h=-(f(x_0)/f(x'_0)$;

we get the (n+1) the corrected root as ,

$$\boxed{x_{n+1}=x_n-\ f(x_n)/f'(x_n)}$$

PROGRAM CODE:

```
//C Program to find the Soln of an Equation using Newton-Raphson's
#include <stdio.h>
#include <math.h>
double eq(double x,int d)
{
      if(!d)
            return (pow(x,3)-x-.1);
      else
            return (3*pow(x,2)-1);
}
double mod(double x)
{
      if(x<0)
            return -x;
      else
            return x;
}
double error(int a)
{
      return 5*pow(10,-a-1);
}
int main()
{
      int i=0;
      double x,e,h,y;
      printf("The equation we are solving is->   x^3-x-.1=0\n");
      printf("Enter value of x0::");
      scanf("%lf",&x);
      y=x;
      printf("You need the answer correct upto how many decimal 
places? ::");
      scanf("%lf",&e);
      e=error(e);
      printf("n\tx(n)\t\t\tf(x(n))\t\t\tf'(x(n))\t\th\t\t\tx(n+1)\n");
      do
      {
            x=y;
```

```
          h=-eq(x,0)/eq(x,1);
          y=x+h;
          printf("%d\t%lf\t\t%lf\t\t%lf\t\t%lf\t\t%lf\n",i+
+,x,eq(x,0),eq(x,1),h,y);
     }
     while(mod(x-y)>e);
     printf("The root is %lf\n",x);
     return 0;
}
```

OUTPUT:

```
The equation we are solving is->x^3-x-.1=0
Enter value of x0::1
You need the answer correct upto how many decimal places? ::6
n     x(n)       f(x(n))          f'(x(n))   h          x(n+1)
0     1.000000   -0.100000        2.000000   0.050000   1.050000
1     1.050000   0.007625         2.307500   -0.003304  1.046696
2     1.046696   0.000034         2.286715   -0.000015  1.046681
3     1.046681   0.000000         2.286620   -0.000000  1.046681
The root is 1.046681
```