

```

1 // Solving Polynomial Equation by using Newton-Rhapson Method
2 // Given Function  $f(x) = x^3 - x^2 + 2$ 
3
4 #include<stdio.h>
5 #include<math.h>
6 #include<stdlib.h>
7 #include<conio.h>
8
9
10 // Function Prototype
11 float F(float x); // Original Function
12 float f(float x); // Differential Coefficient of F(x)
13 float tolerance(float x1, float x2);
14 int NR_method (float x1, float TOL);
15
16 // main() Function
17 void main()
18 {
19     float TOL, x1; //TOL = Desired Tolerance X1,X2=Initial boundary
20     printf("##### This Program is to solve a equation by Newton-Rhapson Method\n\n");
21     printf("Please enter Tolerance : ");
22     scanf("%f",&TOL);
23
24     printf("\nEnter the initial approximation of the solution :: ");
25     scanf("%f", &x1);
26
27     NR_method(x1,TOL);
28
29     getch();
30 }
31
32
33 // Defining function f(x) | F(x) stands for anti-derivative i.e. the given function
34 float F(float x)
35 {
36     float Fx = x*x*x - x*x +2; // Define f(x)
37     return Fx;
38 }
39
40 // Defining Differential Co-efficient
41 float f(float x)
42 {
43     float fx = 3*x*x -4*x; // Define f'(x)
44     return fx;
45 }
46
47 //Defining Tolerance function
48 float tolerance(float x1, float x2)
49 {
50     float TOL = fabs(x2-x1);
51     return TOL;
52 }
53
54
55 //Defining Bisection Method
56 int NR_method (float x1, float TOL)
57 {
58     float x2;
59     int n; //n=number of iteration

```

```
60
61     for(n=1; n>0; n++)
62     {
63         x2 = x1 - F(x1)/f(x1); // p[n] = p[n-1]-f(p[n-1])/f'(p[n-1])
64
65         if (tolerance(x1,x2)<=TOL)
66         {
67             printf("\nSolution of the Polynomial equation is :: %f", x2);
68             printf("\nNumber of Iteration :: %d\n\n", n);
69             break;
70         }
71         else x1=x2;
72     }
73 }
74
```