Newton Ralson Method:

$f(x) = x^3 - 10x + 1$

Guess Values:

x	$f(x)$
-10	-899
-9	-638
-8	-431
-7	-272
-6	-155
-5	-74
-4	-23
-3	4
-2	13
-1	10

0	1
1	-8
2	-11
3	-2
4	25
5	76
6	157
7	274
8	433
9	640
10	901

We take guess value (3, 4)

★ $x_0 = 3$ lower bound

$f(x) = x^3 - 10x + 1$

$f'(x) = 3x^2 - 10$

formula: $x_i - f(x_i) / f'(x_i)$

$x_1 = x_0 - f(x_0) / f'(x_0)$ —— (i)

We have to find

-> $f(x_0)$

-> $f'(x_0)$

★ $f(x_0) = (3)^3 - 10(3) + 1$

$= -2$

★ $f'(x_0) = 3(3)^2 - 10$

$= 17$

By Putting values of x_0, $f(x_0)$ and $f'(x_0)$ in equation (1)

$x_1 = 3 - \left(-2/17 \right) => 3.1176$

★ $x_1 = 3.1176$

$x_2 = x_1 - f(x_1) / f'(x_1)$ —— (ii)

★ $f(x_1) = f(3.1176) = (3.1176)^3 - 10(3.1176) + 1$

$= 30.3012 - 31.176 + 1$

$= 0.1252$

★ $f'(x_1) = f'(3.1176) = 3(3.1176)^2 - 10$

$= 3(9.719) - 10$

$= 19.1582$

By putting value of x_1, $f(x_1)$, $f'(x_1)$ in eqn (ii)

$$x_2 = 3.1176 - \left(0.1252 / 19.1582 \right) \Rightarrow 3.1111$$

★ $x_2 = 3.1111$

$$x_3 = x_2 - f(x_2) / f'(x_0) \quad \text{——— (iii)}$$

★ $f(x_2) = f(3.1111) = (3.1111)^3 - 10(3.1111) + 1$

$= 30.1122 - 31.111 + 1$

$= 0.0012$

★ $f'(x_2) = f'(3.1111) = 3(3.1111)^2 - 10$

$= 3(9.678) - 10$

19.034

Putting value of $f(x_2)$, $f'(x_2)$, x^2 in equation (iii)

$$x_3 = 3.1111 - \left(0.0012 / 19.034 \right)$$

3.1110

★ $x_3 = 3.1110$

$$\cancel{x_4 = x_3 - f(x_3) / f'(x_3)} \quad \cancel{\text{——— (iv)}}$$

★ $\cancel{f(x_3) = (3.0533)^3 - 10(3.0533) + 1}$

$= 28.4648 - 30.533 + 1$

$= -1.0682$

★ $f'(x_3) = 3(3.0533) - 10$

$= 9.1599 - 10$

$= -0.840$

Putting values of x_3, $f(x_3)$, $f'(x_3)$ in equation (IV)

$x_4 = 3.0533 - \left(\frac{+1.0682}{+0.840} \right)$

$= 1.7816$

$x_4 = 1.7816$

$|x_n - x_{n-1}|$

$|x_3 - x_2| = 3.1110 - 3.1111$

$= 0.001$

As the differance between $|x_n - x_{n-1}| < 0.001$

So we stop computing.

PROGRAM:

```
#include<iostream.h>
#include<conio.h>
#include<math.h>

int main()
{
  float x[20], F, FD;
  int i=0;
```

```
Cout<<"Enter the Guess value of lower Bound";
Cin>> x[0];
 do
   {
      F= (x[i])^3 -10 (x[i]) +1
      FD= 3(x[i])^2 -10

      x[i+1] = x[i] - F/FD

      Cout<< x[i];
          i++
   while ( fabs x[i-1 - x[i]) >0.0001
   }
   return 0;
```

Algorithm:

Newton Ralson Method for finding roots where 'x' is array of 20 elements. and i is a loop Counter T, F and FD are variables.

"x" is a lower bound Guess value.
F and FD are functional Value
T= 0.0001

Step 1: Read x[0]
Step 2: Let i = 0
Step 3: Do Step 4 to 8 while((x[i-1] - x[i]) >T)

Step 4: F = (x[i]*x[i]*x[i] - 10*(x[i]) + 1

Step 5: FD = 3 * (x[i] * x[i]) - 10

Step 6: x[i + 1] = x[i] - F/FD

Step 7: write x[i]

Step 8: increment in i

Step 9: write "Root is" x[i]