

North South University

Department of Electrical and Computer Engineering

Assignment -2

Are you kidding Mr. Feynman

Name : Joy Kumar Ghosh

Student ID : 2211424 6 42

Course No : CSE 115

Course Title: Programming Language I

Section : 3

Date : 2 April, 2022

1

Given that,

the equation of square root is,

$$\sqrt{n} = a + dx$$

Here, $n = \text{number}$

$a = \text{nearest square root of a perfect square.}$

We can use this equation for calculating cubic root. Like,

$$\sqrt[3]{n} = a + dx \dots (i)$$

Here,

$n = \text{number}$

$a = \text{nearest cubic root of a perfect}$

cube ; perfect cube must less than

or equal to the number.

Now,

$$\sqrt[3]{n} = a + dx$$

$$\Rightarrow n = (a + dx)^3$$

$$= a^3 + 3 \cdot a^2 \cdot dx + 3 \cdot a \cdot (dx)^2 + (dx)^3$$

Since this is only an approximated method, dx is a very small area; if we multiply it twice or three times, it will be smaller than dx .

So, we can ignore these small areas ($3 \cdot a \cdot dx$ and $(dx)^2$).

So, the equation will be,

$$n = a^3 + 3 \cdot a^2 \cdot dx$$

$$\Rightarrow n - a^3 = 3 \cdot a^2 \cdot dx$$

$$\therefore dx = \frac{n - a^3}{3 \cdot a^2}$$

Now, we get the value of dx . By substituting dx in the above equation (i), we can calculate the root.

Finally, we can help Feynman.

21

The complete program in C is attached by taking a screenshot so that all the comments and code syntax remain clear to read.


```
2  /* Writing a c program to calculate cubic root in analog procedure.
3  */
4
5  #include <stdio.h>
6
7  int main()
8  {
9      double number, dx, cubic_root, a, nearest_perfect_cube;
10
11     //this loop is infinity it will take input until get zero as input
12     while(1){
13         printf("Enter a Number: ");
14         scanf("%lf", &number);
15
16         if(number == 0){//this condition for exit the program
17             printf("\nExit Successfully.\n");//simple exit message
18             break;//if user give 0 as input it will break the loop
19         }
20
21         //validity check loop. this loop will check the range of input
22         while(number < 1 || number > 1000000){
23             printf("\nOut of Range Enter again: ");
24             scanf("%lf", &number);
25             if(number == 0){
26                 break;
27             }
28         }
29
30         //again check exit command as i found an error in previous loop.
31         if(number == 0){//this condition for exit the program
32             printf("\nExit Successfully.\n");//simple exit message
33             break;//if user give 0 as input it will break the loop
34         }
35
36         //this loop will calculate the nearest perfect cube
37         for(a = 1, nearest_perfect_cube = 1; nearest_perfect_cube <= number; a++){
38             nearest_perfect_cube = a * a * a;
39         }
40     }
```

```
41     a = a - 2; //as nearest perfect cube increment the value of a twice up, so decrease it by 2. so we can find the nearest perfect cubic root accurately
42
43     dx = (number - (a * a * a)) / (3 * a * a); //this statement is for calculating the dx part
44
45     cubic_root = a + dx; //main cubic root rules
46
47     printf("Cubic Root of %0.4lf is %0.4lf\n\n", number, cubic_root); //printing the cubic root
48 }
49
50 return 0;
51 }
52
```