

Student Name: ..... **KEY SOLUTION** ..... ID: .....

The  $n^{\text{th}}$  root of a number  $X$  is computed using the iterative formula:

$$x_{k+1} = \frac{1}{n} \left[ (n-1)x_k + \frac{X}{x_k^{n-1}} \right]$$

The iterations will stop when the difference between two (2) successive terms is less than a small value  $\epsilon$  i.e.  $|x_{k+1} - x_k| \leq \epsilon$ , where  $\epsilon$  is a constant value equal to  $10^{-5}$ . Assume the initial value  $x_0 = X/n$ ,  $x_k$  the old value and  $x_{k+1}$  the updated or new value.

Write a complete C program that prompts the user to enter the values of  $X$  and  $n$ , then computes the  $n^{\text{th}}$  root of  $X$ .

- The above formula must be implemented as a function **rootN()** which accepts both  $n$  and  $x$  as inputs and returns the  $n^{\text{th}}$  root of  $x$ .
- The code should check that when  **$n$  is even**, the value of  **$X$**  has to be **positive**, and for that purpose you need another small function that returns **TRUE (1)** if  $n$  is even and **FALSE** otherwise.

```
Enter n: 4
Enter X: -8
Wrong Input. Repeat again !
Enter n: 4
Enter X: 8
(4)th root of (8.00) = 1.681793
```

```
// Quiz 3 Sec 51 T 191
```

```
#include <stdio.h>
```

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

```
#define EPS 1E-5
```

```
double rootN(double x, int n);
```

```
int odd(int a);
```

```
int main(){
```

```
    int n;
```

```
    double X;
```

```
    do{
```

```
        printf("Enter n: ");
```

```
        scanf("%d", &n);
```

```
        printf("Enter X: ");
```

```
        scanf("%lf", &X);
```

```
        if(!odd(n) && (X < 0))
```

```
            printf("Wrong Input. Repeat again !\n");
```

```
    }while(!odd(n) && (X < 0));
```

```
    printf("( %d)th root of ( %0.2f) = %0.6f\n", n,X,rootN(X,n));
```

```
    return 0;
```

```
}
```

```
double rootN(double X,int n){
```

```
    double error, xo, xn;
```

```
    xo = X/n;
```

```
    do{
```

```
        xn = ((n-1)*xo+X/pow(xo,n-1))/n;
```

```
        error = fabs(xo-xn);
```

```
        xo = xn;
```

```
    }while(error > EPS);
```

```
    return(xo);
```

```
}
```

```
int odd(int a){
```

```
    return(a%2);
```

```
}
```