

PROGRAM: 4

Name: Kiran Kanyal

Roll No: 29

Section: A1

Aim: Write a program in c language to find the roots of a given algebraic equation using the Secant Method.

ALGORITHM:

START

1. Prompt the user to input two initial guesses x_0 and x_1 .
2. Verify that $f(x_0) \cdot f(x_1) < 0$ (indicating that root between them). If not, prompt the user to try again.
3. Prompt the user to input the solution's allowed error (allowederr).
4. **Step 1:** Calculate the next approximation x_2 using the Secant Method formula: $x_2 = \frac{(x_1 - x_0) * f(x_1)}{f(x_1) - f(x_0)}$;
5. **Step 2:** Check if $fabs(x_1 - x_2)$ is less than or equal to the allowed error:
 - a. If yes, the root is found, and the process stops.
 - b. If no, update $x_0 = x_1$ and $x_1 = x_2$, then repeat the iteration process
6. **Output:** If the root is found within the allowed error, display the root and the number of iterations. If not, print the current iteration number, and the values of x_0 , x_1 , and x_2 , and repeat the process.

STOP

PROGRAM:

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

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

```
float f(float x) {  
    return (x * x * x - 5 * x + 1); //return pow(x, 2) * exp(-x / 2) - 1;  
}
```

```

float g(float x0, float x1) {
    float x2 = x1 - ((x1 - x0) * f(x1)) / (f(x1) - f(x0));
    return x2;
}

int main() {
    int flag = 0, iter = 0;
    float x0, x1, allowederr;

    do {
        printf("Enter the initial guess of x0 and x1:\n");
        scanf("%f%f", &x0, &x1);

        if (f(x0) * f(x1) < 0)
        {
            // Check if a root exists between x0 and x1
            flag = 1;
            printf("Initial guess accepted.\n");
        }
    else {
        printf("Wrong initial guess, f(x0)* f(x1) = %f is not less than 0. Try again.\n",
            fabs(f(x0)) * fabs(f(x1)));
    }
    } while (flag != 1);

    printf("Enter allowed error:\n");
    scanf("%f", &allowederr);

    do {
        iter++;
        float x2 = g(x0, x1);

```

```
    printf("%d iteration, value of x0 is %f, value of x1 is %f and g(x0, x1) is %f\n", iter, x0,  
x1, x2);
```

```
    if (fabs(x1 - x2) <= allowederr) {
```

```
        flag = 0;
```

```
        printf("Root of the equation is %f after %d iterations.\n", x2, iter);
```

```
        break;
```

```
    }
```

```
    else {
```

```
        x0 = x1;
```

```
        x1 = x2;
```

```
    }
```

```
    } while (flag != 0);
```

```
    return 0;
```

```
}
```

OUTPUT:

For equation: $f(x) = x^3 - 5x + 1 = 0$

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - KIRAN KANYAL ROLL NO = 29 SECTION = A1 + v [] [] ... ^

● PS C:\Users\admin\Desktop\CBNST> cd "c:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1\" ; if ($?) { gcc Pr
ogram4_1.c -o Program4_1 } ; if ($?) { .\Program4_1 }
Enter the initial guess of x0 and x1:
1 2
Wrong initial guess, f(x0)* f(x1) = 3.000000 is not less than 0. Try again.
Enter the initial guess of x0 and x1:
0 1
Initial guess accepted.
Enter allowed error:
0.0001
1 iteration, value of x0 is 0.000000, value of x1 is 1.000000 and g(x0, x1) is 0.250000
2 iteration, value of x0 is 1.000000, value of x1 is 0.250000 and g(x0, x1) is 0.186441
3 iteration, value of x0 is 0.250000, value of x1 is 0.186441 and g(x0, x1) is 0.201736
4 iteration, value of x0 is 0.186441, value of x1 is 0.201736 and g(x0, x1) is 0.201640
Root of the equation is 0.201640 after 4 iterations.
○ PS C:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1>
```

For equation: $f(x) = (x^2) * (e^{(-x/2)}) = 1$

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - KIRAN KANYAL ROLL NO = 29 SECTION = A1 + v [] [] .

● PS C:\Users\admin\Desktop\CBNST> cd "c:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1\" ; if ($?) { gcc Pr
ogram4_2.c -o Program4_2 } ; if ($?) { .\Program4_2 }
Enter the initial guess of x0 and x1:
0 1
Wrong initial guess, f(x0)* f(x1) = 0.393469 is not less than 0. Try again.
Enter the initial guess of x0 and x1:
0 2
Initial guess accepted.
Enter allowed error:
0.0001
1 iteration, value of x0 is 0.000000, value of x1 is 2.000000 and g(x0, x1) is 1.359141
2 iteration, value of x0 is 2.000000, value of x1 is 1.359141 and g(x0, x1) is 1.435459
3 iteration, value of x0 is 1.359141, value of x1 is 1.435459 and g(x0, x1) is 1.429648
4 iteration, value of x0 is 1.435459, value of x1 is 1.429648 and g(x0, x1) is 1.429612
Root of the equation is 1.429612 after 4 iterations.
○ PS C:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1>
```

PROGRAM: 5

Name: Kiran Kanyal

Roll No: 29

Section: A1

Aim: Write a program in c language to find the roots of a given algebraic equation using Newton Raphson Method.

ALGORITHM:

START

1. Prompt the user to enter an initial guess (x0).
2. Prompt the user to input the solution's allowed error (allowederr).
3. Initialize an iteration counter (iter) to 0.
4. Repeat the following steps until convergence:
 - a. **Step 1:** Increment the iteration counter (iter).
 - b. **Step 2:** Compute the next approximation using the Newton-Raphson formula:
$$x1 = x0 - f(x0) / f'(x0)$$
 - c. **Step 3:** Print the current iteration number, x0, and x1.
 - d. **Step 4:** Check for convergence:
 - i. If $\text{fabs } |x0 - x1| \leq \text{allowederr}$
 1. Print the root found (x1) and the number of iterations.
 2. Exit the loop.
 - e. **Step 5:** Update x0 to x1 for the next iteration.
5. Once the root is found within the allowed error, terminate the program.

STOP

PROGRAM:

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

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

```

float f(float x) {
    return 3*x - cos(x) - 1;
    // return x*x- 12;
}

float df(float x) { // derivative of f(x)
    return 3 + sin (x);
    //return 2*x;
}

int main() {
    int iter = 0;
    float x0, x1, allowederr;

    printf("Enter the initial guess:\n");
    scanf("%f", &x0);
    printf("Enter allowed error:\n");
    scanf("%f", &allowederr);

    do {
        iter++;
        x1 = x0 - f(x0) / df(x0); //Newton Raphson Formula

        printf("%d iteration, x0 = %f, x1 = %f, \n", iter, x0, x1);

        if (fabs(x0 - x1) <= allowederr) {
            printf("Root of the equation is %f after %d iterations.\n", x1, iter);
            break;
        }
        x0 = x1;
    } while (1);
    return 0;
}

```

OUTPUT:

For equation: $3x - \cos x - 1 = 0$



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

 Code - KIRAN KANYAL ROLL NO = 29 SECTIC

```
● PS C:\Users\admin\Desktop\CBNST> cd "c:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1\'  
goram5_1.c -o Prgoram5_1 } ; if ($?) { .\Prgoram5_1 }  
Enter the initial guess:  
0.6  
Enter allowed error:  
0.0001  
1 iteration, x0 = 0.600000, x1 = 0.607107,  
2 iteration, x0 = 0.607107, x1 = 0.607102,  
Root of the equation is 0.607102 after 2 iterations.  
○ PS C:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1> █
```

For equation: $x^2 - 12 = 0$

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

 Code - KIRAN KANYAL ROLL NO = 29 SECTION = A1 + ▾ 

```
● PS C:\Users\admin\Desktop\CBNST> cd "c:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1\" ; if ($?) { {  
goram5_1.c -o Prgoram5_1 } ; if ($?) { .\Prgoram5_1 }  
Enter the initial guess:  
3.2  
Enter allowed error:  
0.0001  
1 iteration, x0 = 3.200000, x1 = 3.475000,  
2 iteration, x0 = 3.475000, x1 = 3.464119,  
3 iteration, x0 = 3.464119, x1 = 3.464102,  
Root of the equation is 3.464102 after 3 iterations.  
○ PS C:\Users\admin\Desktop\CBNST\KIRAN KANYAL ROLL NO = 29 SECTION = A1> █
```