**Regula Falsi-False position Method:**

***PSEUDOCODE:***1. Start

2. Define function f(x)

3. Input

a. Lower and Upper guesses x0 and x1

b. tolerable error e

4. If f(x0)\*f(x1) > 0

print "Incorrect initial guesses"

goto 3

End If

5. Do

x2 = x0 - ((x0-x1) \* f(x0))/(f(x0) - f(x1))

If f(x0)\*f(x2) < 0

x1 = x2

Else

x0 = x2

End If

While abs(f(x2) > e

6. Print root as x2

7. Stop

***Algorithm:***

1. start

2. Define function f(x)

3. Choose initial guesses x0 and x1 such that f(x0)f(x1) < 0

4. Choose pre-specified tolerable error e.

5. Calculate new approximated root as:

x2 = x0 - ((x0-x1) \* f(x0))/(f(x0) - f(x1))

6. Calculate f(x0)f(x2)

a. if f(x0)f(x2) < 0 then x0 = x0 and x1 = x2

b. if f(x0)f(x2) > 0 then x0 = x2 and x1 = x1

c. if f(x0)f(x2) = 0 then goto (8)

7. if |f(x2)|>e then goto (5) otherwise goto (8)

8. Display x2 as root.

9. Stop

**Secant Method**

***Pseudocode:***

1. Start

2. Define function as f(x)

3. Input:

a. Initial guess x0, x1

b. Tolerable Error e

c. Maximum Iteration N

4. Initialize iteration counter step = 1

5. Do

If f(x0) = f(x1)

Print "Mathematical Error"

Stop

End If

x2 = x1 - (x1 - x0) \* f(x1) / ( f(x1) - f(x0) )

x0 = x1

x1 = x2

step = step + 1

If step > N

Print "Not Convergent"

Stop

End If

While abs f(x2) > e

6. Print root as x2

7. Stop

***Algorithm:***

1. Start

2. Define function as f(x)

3. Input initial guesses (x0 and x1),

tolerable error (e) and maximum iteration (N)

4. Initialize iteration counter i = 1

5. If f(x0) = f(x1) then print "Mathematical Error"

and goto (11) otherwise goto (6)

6. Calcualte x2 = x1 - (x1-x0) \* f(x1) / ( f(x1) - f(x0) )

7. Increment iteration counter i = i + 1

8. If i >= N then print "Not Convergent"

and goto (11) otherwise goto (9)

9. If |f(x2)| > e then set x0 = x1, x1 = x2

and goto (5) otherwise goto (10)

10. Print root as x2

11. Stop