Mesuga, Reymond R.
BS Physics 3-1
Computational Physics 1

```
# Importing necessary modules and package
import numpy as np
from numpy import cos
from numpy import sin
from numpy import exp
from numpy import log
```

```
# Defining the function that can solve for roots using Bisection Method


def bisection(f,a,b,N):

    # Where f is the given equation
    # a and b is the given interval
    # N is the number of iterations



    if f(a)*f(b) >= 0:
        print("Bisection method fails.")
        return None
    a_n = a
    b_n = b
    for n in range(1,N+1):
        m_n = (a_n + b_n)/2
        f_m_n = f(m_n)
        if f(a_n)*f_m_n < 0:
            a_n = a_n
            b_n = m_n
        elif f(b_n)*f_m_n < 0:
            a_n = m_n
            b_n = b_n
        elif f_m_n == 0:
            print("Found exact solution.")
            return m_n
        else:
            print("Bisection method fails.")
            return None
    return (a_n + b_n)/2
```

```
# The examples below are existing equations that has difined roots already

f = lambda x: x**2 - x - 1
approx_phi = bisection(f,1,2,25)
print(approx_phi)
```

```
1.618033990263939
```

```
y = lambda x: 3*x + sin(x) - exp(x)
approx = bisection(y,0,0.5,6)
print(approx)
```

```
0.36328125
```

```
y1 = lambda x: exp(x)*log(x) -x**2
approx1 = bisection(y1,1,2,20)
print(approx1)
```

```
1.6946005821228027
```