[64]:
```
def roots(f,a,b,N):
    roots=[] #this is an empty list
    if f(a)*f(b) >= 0:
            print("Not looking for roots for functions with a and b both being the same sign")
            return None
    a=a
    b=b
    for n in range(1,N+1):
        M=(a+b)/2
        if f(a) * f(M) < 0:
            b=M
            a=a
        elif f(b) * f(M) < 0:
            a=M
            b=b
    return(a+b)/2
```

[78]:
```
import math
import numpy as np
f = lambda x: math.exp(x) + np.log(x)
approx_phi = roots(f,0,1,9)
print(approx_phi)
```

```
0.2705078125
```

]:
```
import math
f = lambda x: math.atan(x) - x**2
approx_phi = roots(f,0,2,9)
print(approx_phi)
```

```
Not looking for roots for functions with a and b both being the same sign
None
```

]:
```
import math
import numpy as np
f = lambda x: math.sin(x)/ np.log(x)
approx_phi = roots(f,3,4,9)
print(approx_phi)
```

```
3.1416015625
```

]:
```
import math
import numpy as np
f = lambda x: np.log(math.cos(x))
approx_phi = roots(f,5,7,9)
print(approx_phi)
```

```
Not looking for roots for functions with a and b both being the same sign
None
```