Section 2.1

```
import math

def targetfunc(x):
    return 2 * x * math.cos(2*x) - (x+1) * (x+1)

def bisection(a,b):

    if (targetfunc(a) * targetfunc(b) >= 0):
        print("Invalid interval!")
        return

    p = a
    while ((b-a) >= 0.00001):

        # Find middle point
        p = (a+b)/2

        # Check if middle point is root
        if (targetfunc(p) == 0.0):
            break

        # Decide the side to repeat the steps
        if (targetfunc(p)*targetfunc(a) < 0):
            b = p
        else:
            a = p

    print("The value of root is : ")
    print(p)



# question 5(c)
a_1 =-3
b_1 =-2
bisection(a_1, b_1)
```

```
The value of root is :
-2.1913070678710938
```

```
#question 5(c)
a_2 = -1
b_2 = 0
bisection(a_2, b_2)
```

```
The value of root is :
-0.7981643676757812
```

```
import math
```

```
def targetfunc(x):
    return x * math.cos(x) - 2*x*x +3*x -1

def bisection(a,b):

    if (targetfunc(a) * targetfunc(b) >= 0):
        print("Invalid interval!")
        return

    p = a
    while ((b-a) >= 0.00001):

        # Find middle point
        p = (a+b)/2

        # Check if middle point is root
        if (targetfunc(p) == 0.0):
            break

        # Decide the side to repeat the steps
        if (targetfunc(p)*targetfunc(a) < 0):
            b = p
        else:
            a = p

    print("The value of root is : ")
    print(p)


# Question 5(d)
a_3 = 0.2
b_3 = 0.3
bisection(a_3, b_3)
```

```
The value of root is :
0.29752807617187504
```

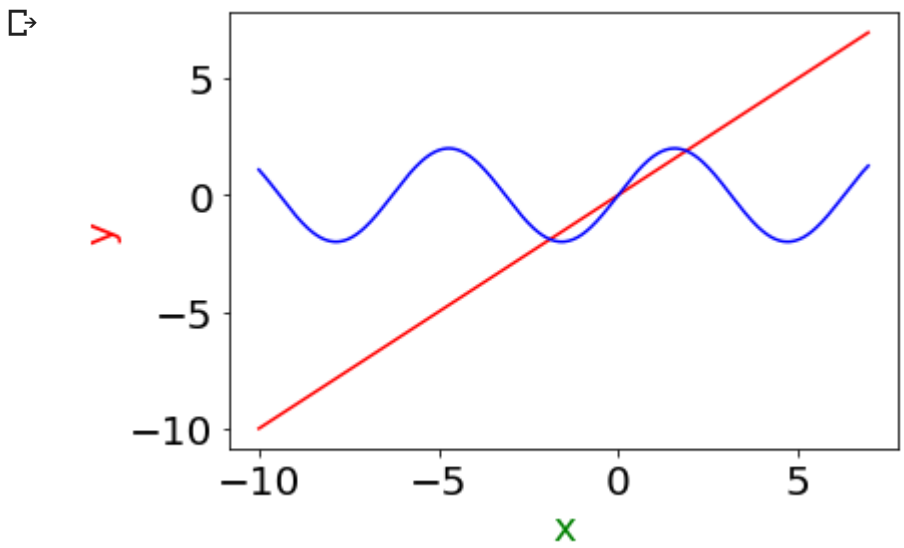
```
# Question 5(d)
a_4 = 1.2
b_4 = 1.3
bisection(a_4, b_4)
```

```
The value of root is :
1.256622314453125
```

```
#Question 7(a)
import matplotlib.pyplot as plt
import numpy as np
import matplotlib

matplotlib.rcParams['font.size']=20
a=np.arange(-10.0,7.0,0.04)
plt.plot(a,a,'r-')
plt.plot(a, 2*np.sin(a), 'b-')
```

```
plt.xlabel('x',color='green')
plt.ylabel('y',color='red')
plt.show()
```




```
# Question 7(b)
import math

def targetfunc(x):
    return x - 2*math.sin(x);

def bisection(a,b):

    if (targetfunc(a) * targetfunc(b) >= 0):
        print("Invalid interval!")
        return

    p = a
    while ((b-a) >= 0.00001):

        # Find middle point
        p = (a+b)/2

        # Check if middle point is root
        if (targetfunc(p) == 0.0):
            break

        # Decide the side to repeat the steps
        if (targetfunc(p)*targetfunc(a) < 0):
            b = p
        else:
            a = p

    print("The value of root is : ")
    print(p)
a_5 = 1.5
b_5 = 2.0
bisection(a_5, b_5)
```

```
The value of root is :
1.8955001831054688
```

Section 2.2

```
# Question 10
def g(x):
    return math.pow(2, -x)

def fixedPoint(p0, tol):
    ite = 1
    while (1):
        p = g(p0)
        if abs(p-p0) < tol:
            break;
        p0 = p
        ite +=1;
    return p,ite

fixedPoint(1/3,0.0001)
```

```
(0.6411661053686456, 12)
```

The approximation of the fixed point accurate to within 10^-4 is 0.6411661053686456

```
# Question 14(a)
def g(x):
    return 2 + math.sin(x)

def fixedPoint(p0, tol):
    ite = 1
    while (1):
        p = g(p0)
        if abs(p-p0) < tol:
            break;
        p0 = p
        ite +=1;
    return p,ite

fixedPoint(2.5,0.00001)
```

```
(2.5541921027478667, 52)
```

```
# Question 14(c)
def g(x):
    return math.sqrt(math.exp(x)/3)

def fixedPoint(p0, tol):
    ite = 1
    while (1):
        p = g(p0)
        if abs(p-p0) < tol:
```

```
            if abs(p-p0) < tol:
                break;
            p0 = p
            ite +=1;
        return p,ite

fixedPoint(1/2,0.00001)
```

```
(0.9100019674023, 14)
```