In [ ]:
```
def fx(x):
    return (1/(1+x**2))
```

In [ ]:
```
def differenceH(upperLim, LowerLim, intervals):
    loop_cond = True
    H =[0]
    difference = (upperLim - LowerLim )/intervals
    a = difference

    while loop_cond is True:
        H.append(difference)

        difference = difference + a
        #print(difference)


        if (difference / upperLim) > 1:
            loop_cond = False

    return H
```

In [ ]:
```
H=[]
H = differenceH(1,0,5)
print(H)
```

```
[0, 0.2, 0.4, 0.6000000000000001, 0.8, 1.0]
```

In [ ]:
```
approx =[]
for i in range(len(H)):

    approx.append(fx(H[i]))
    print(approx[i])
```

```
1.0
0.9615384615384615
0.8620689655172413
0.7352941176470588
0.6097560975609756
0.5
```

In [ ]:
```
(0.2/2)* (((approx[0] + approx[]) + (2*sum(approx[1:-1]))))   #H == 0.2
```

Out[ ]:
```
0.7837315284527475
```

In [ ]:
```
approx[:1]
```

Out[ ]:
```
[1.0]
```

Question 2

In [ ]:
```
import scipy.integrate as si
import math as m

#ans,err = si.quad(fx,0,1)
#ans
```

In [ ]:
```
def fx1(t):
    return(2000*m.log(140000/ (140000 - 2100*t)) - 9.8*t)
```

In [ ]:
```
ans,err = si.quad(fx1,8,30)
ans
```

Out[ ]:
```
11061.335535080994
```

In [ ]:
```
arr =[]
arr = differenceH(30,8,2)
```

In [ ]:
```
arr2 =[]
for i in range(len(arr)):

    arr.append(fx(arr[i]))
    print(arr[i])
```

```
0
11.0
22.0
```