

for #4-7

```
import random
import math
```

#4

```
def calculateA():
    x1 = random.uniform(-1, 1)
    x2 = random.uniform(-1, 1)
    y1 = math.sin(math.pi * x1)
    y2 = math.sin(math.pi * x2)
    a = (x1*y1 + x2*y2) / (x1**2 + x2**2)
    return a
```

#5

```
def calculateBias(aHat):
    x = random.uniform(-1, 1)
    y = math.sin(math.pi * x)
    gBarx = aHat * x
    return (gBarx - y)**2
```

#6

```
def calculateVar(aHat):
    #get a for  $g^{(D)}(x)$ 
    a = calculateA()
    totalVar = 0
    # multiple x values to get  $E_x$ 
    for i in range(0, 1000):
        x = random.uniform(-1, 1)
        gx = a * x
        gBarx = aHat * x
        totalVar += (gx - gBarx)**2
    return totalVar / 1000
```

#7 $h = ax^2$

```
def calculateE0Out():
    #get a for  $g^{(D)}(x)$ 
    x1 = random.uniform(-1, 1)
    x2 = random.uniform(-1, 1)
    y1 = math.sin(math.pi * x1)
    y2 = math.sin(math.pi * x2)
    a = ((x1**2)*y1 + (x2**2)*y2) / (x1**4 + x2**4)
    totalE0Out = 0
    # multiple x values to get  $E_x$ 
    for i in range(0, 1000):
        x = random.uniform(-1, 1)
        gx = a * (x**2)
```

```

        y = math.sin(math.pi * x)
        totalE0ut += (gx - y)**2
    return totalE0ut / 1000

```

```

#7 h = ax^2 + b
def calculateE0ut2():
    x1 = random.uniform(-1, 1)
    x2 = random.uniform(-1, 1)
    y1 = math.sin(math.pi * x1)
    y2 = math.sin(math.pi * x2)
    a = (y1 - y2) / (x1**2 - x2**2)
    b = y1 - a * (x1**2)
    totalE0ut = 0
    # multiple x values to get E_x
    for i in range(0, 1000):
        x = random.uniform(-1, 1)
        gx = a * (x**2) + b
        y = math.sin(math.pi * x)
        totalE0ut += (gx - y)**2
    return totalE0ut / 1000

```

```

#4 calculate a hat
totalA = 0
for i in range(0, 10000):
    totalA += calculateA()
aHat = totalA / 10000
print(aHat)

```

```

#5 calculate bias
totalBias = 0
for i in range(0, 10000):
    totalBias += calculateBias(aHat)
print(totalBias / 10000)

```

```

#6 calculate variance for a single data set
# has to be repeated for many data sets (different g^(D)(x)) for E_D
totalVar = 0
for i in range(0, 1000):
    totalVar += calculateVar(aHat)
print(totalVar / 1000)

```

```

# 7 calculate E0ut for a single data set
# has to be repeated for many data sets (different g^(D)(x)) for E_D
# for h = ax^2 : calculateE0ut()
# for h = ax^2 + b : calculateE0ut2()
totalE0ut = 0
for i in range(0, 1000):
    totalE0ut += calculateE0ut2()

```

```
print(totalEOut / 1000)
```