Caitlin Sisilli

Exam 1

Question 1)

A)

```
keys=["a","b","c","d","e","f"]

values=[0,1,2,3,4,5]

diction= dict(zip(keys,values))

print(diction)
```

Output:

```
{'a': 0, 'b': 1, 'c': 2, 'd': 3, 'e': 4, 'f': 5}
```

B)

a) True

b) False

c) False

d) True

e) True

Question 2)

A)

```
Import math

Def function(x):

    F=2 * math.sin(x) * math.cos(x)

    Return F

Def function_f(x):

    Two_f=2 * (math.cos(x) * math.cos(x) -  math.sin(x) * math.sin(x))

    Return Two_f

Def function_ff(x):

    Three_f= -4 * 2 * math.sin(x) * math.cos(x)

    Return Three_f
```

If x is equal to pi like x=pi

The output would be:

0 , 2 , 0

B)

When writing the code, need to separate down all square components.

Separating the x’s from the polynomial function

```
def polynomial(x):
    return x*(x*(x*(x+2)-25)-26)+120
```

Question 3)

A) The output for import numpy as np B = np.arange(12).reshape((3, 4)) is:

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

B)

```
import numpy as np
B = np.arange(12).reshape((3, 4))
sum_col=B.sum(axis=0)
print(sum_col)
```

Output:

```
[12 15 18 21]
```

C)

```
import numpy as np
B = np.arange(12).reshape((3, 4))
min_row=B.min(axis=1)
print(min_row)
```

Output:

```
[0 4 8]
```

Question 4)

First:

Np.arrange arrange 0-40, as well as take the argument. It then will generate argument 10 between the start of 0 through 40 which A will look like: [0,10,20,30]

Np.tile will broadcast and create a new axes by taking the parameter [A,r] r is number of replications.

So the output at the moment is:

[[0,10,20,30],

[0,10,20,30],

[0,10,20,30]]

Then np.T makes a matrixs, where it changes the rows and columns for A to be:

[[0,0,0],

[10,10,10],

[20,20,20],

[30,30,30]])

Then into B=np.array([0,1,2]) . It using different dimension that A and B do not match for, so it call for broadcasting

Output of that be:

*B[[0,1,2],

[0,1,2],

[0,1,2]]

Then last A -B:

[[0,-1,-2],

[10,9,8],

[20,19,18],

[30,29,28]]

Question 5)

A-D)

```
class Numbers:
    MULTIPLIER=2
    def __init__(self,x,y):
        self.x=x
        self.y=y
```

```
    def add(self):
        return self.x+self.y
    def multiply(self,w):
        return self.MULTIPLIER*w
    def difference(self,a,b):
        return a-b
    def value(self):
        list=[self.x,self.y]
        return tuple(i for i in list)
```

Question 6)

A)

It would print out:

Line 1-5:

Letter # 0 is S

Letter # 1 is n

Letter # 2 is o

Letter # 3 is w

Letter # 4 is !

B)

The output would be:

Line 1-4:

X = 4

X = 3

Y = 2

Z = 4