In [92]:

```
import numpy as np
import pandas as pd
from numpy import random,dot
```

In [93]:

```
x1 = 0.1
x2 = 0.4
target = 0.7
learning_rate = 0.01
```

In [94]:

```
weights = random.rand(2)
print(weights)
```

```
[0.52250264 0.76443458]
```

In [95]:

```
def sigmoid(x):
    return 1.0/(1+np.exp(-1*x))
```

In [96]:

```
def error(target,predicted):
    return np.power(target-predicted,2)
```

In [97]:

```
def error_predicted_deriv(predicted, target):
    return 2*(predicted-target)
```

In [98]:

```
def sigmoid_sop_deriv(sop):
    return sigmoid(sop)*(1.0-sigmoid(sop))
```

In [99]:

```
def sop_w_deriv(x):
    return x
```

In [100]:

```
def update_w(w, grad, learning_rate):
    return w - learning_rate*grad
```

In [101]:

```
inputs=[x1,x2]
predicted_outputs=[]
total_error =[]
```

In [102]:

```
for i in range(100):
    print("Epoch ",i," :\n")
    result = dot(weights,inputs)
    print("\tX",result)
    predicted = sigmoid(result)
    print("\t predicted",predicted)
    err = error(target,predicted)
    print("\t error",err)
    predicted_outputs.append(predicted)
    total_error.append(err)
    g1 = error_predicted_deriv(predicted, target)
    g2 = sigmoid_sop_deriv(result)
    g3w1 = sop_w_deriv(x1)
    g3w2 = sop_w_deriv(x2)
    gradw1 = g3w1*g2*g1
    gradw2 = g3w2*g2*g1
    w1 = update_w(w1, gradw1, learning_rate)
    w2 = update_w(w2, gradw2, learning_rate)
    weights =[w1,w2]
    print("\t updated weights:", weights,"\n")
```

```
        X 0.31413745215979805
         predicted 0.5778948436739683
         error 0.014909669201404638
         updated weights: [0.4244223577515736, 0.6794912168872752]


Epoch  98  :

        X 0.31423872253006746
         predicted 0.5779195466028727
         error 0.01490363710164816
         updated weights: [0.42448191556613246, 0.6797294481455106]

Epoch  99  :

        X 0.31433997081481746
         predicted 0.5779442437547179
         error 0.014897607632607718
         updated weights: [0.4245414603923299, 0.6799676274503003]
```

In [103]:

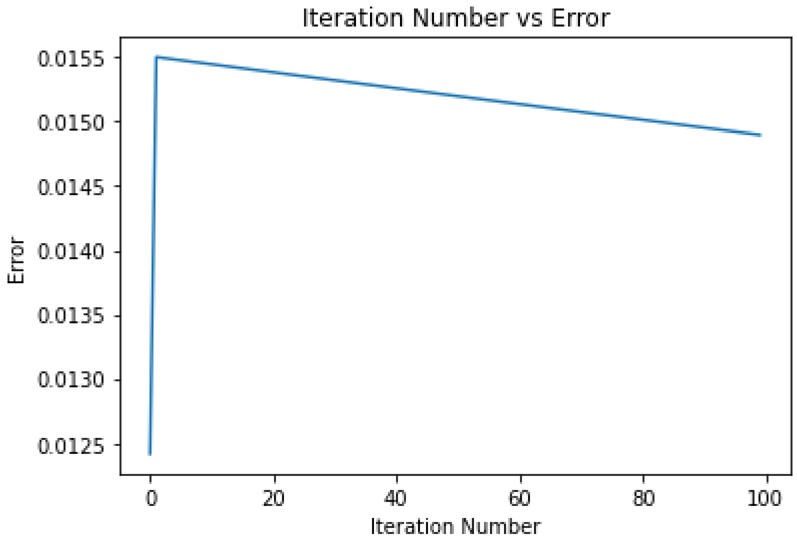
```
import matplotlib.pyplot
```

In [104]:

```
matplotlib.pyplot.figure()
matplotlib.pyplot.plot(total_error)
matplotlib.pyplot.title("Iteration Number vs Error")
matplotlib.pyplot.xlabel("Iteration Number")
matplotlib.pyplot.ylabel("Error")
```

Out[104]:

```
Text(0, 0.5, 'Error')
```

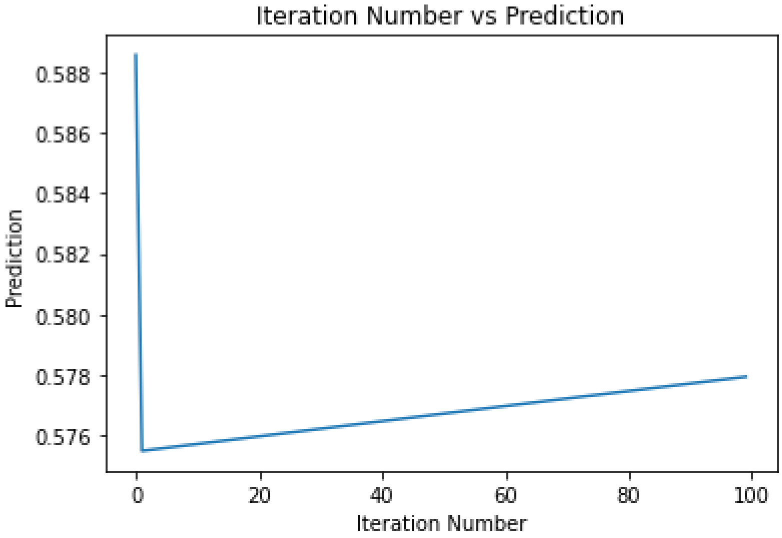


In [105]:

```
matplotlib.pyplot.figure()
matplotlib.pyplot.plot(predicted_outputs)
matplotlib.pyplot.title("Iteration Number vs Prediction")
matplotlib.pyplot.xlabel("Iteration Number")
matplotlib.pyplot.ylabel("Prediction")
```

Out[105]:

```
Text(0, 0.5, 'Prediction')
```



In [ ]:

```

```