In [89]:
```
import numpy as np
import matplotlib.pyplot as pyp
x = np.load("assignment8_X.npy")
y = np.load("assignment8_Y.npy")
x = x.T
y = y.T

alpha = 0.01
n = x.shape[0] # n = 10
t = x.shape[1] # t = 25
#w1 = np.ones((m,n))*0.01
#w2 = np.ones((n,m))*0.01

print(n,t)

def NN1_forward_pass(x,y,w1,w2,w3):
    #print("before forward pass, x")
    #print(x)
    #print("w2")
    #print(w2)
    #print("w1")
    #print(w1)
    #print("y")
    #print(y)
    fwx = w3.dot(w2.dot(w1.dot(x)))
    #print("fwx")
    #print(fwx)
    loss = fwx - y
    #print("loss")
    #print(loss)
    return loss

def NN1_backprop(x,w1,w2,w3,n,m):
    dfdw3 = np.zeros((n,n,m))  # 10x10x25
    dfdw2 = np.zeros((n,m,m))  # 10x25x25
    dfdw1 = np.zeros((n,m,n))  # 10x25x10
    dfdh1 = w3.dot(w2)         # 10x25
    dfdh2 = w3                 # 10x25
    dh1dw1 = np.zeros((m,m,n)) # 25x25x10
    dh2dw2 = np.zeros((m,m,m)) # 25x25x25
    h1 = w1.dot(x)             # 25x1
    h2 = w2.dot(h1)            # 25x1
    #print("In BP, h1", x.shape)
    for i in range(n):
        dfdw3[i][i] = h2
    for i in range(m):
        dh1dw1[i][i] = x
        dh2dw2[i][i] = h1
    for i in range(n):
            for j in range(m):
                dfdw2[i] += dh2dw2[j]*dfdh2[i][j]
                dfdw1[i] += dh1dw1[j]*dfdh1[i][j]
    return dfdw1,dfdw2,dfdw3

def VecXTen(vec,tensor):
    result = np.zeros((tensor.shape[1],tensor.shape[2]))
```

```
    for i in range(len(vec)):
        result += vec[i]*tensor[i]
    return result
```

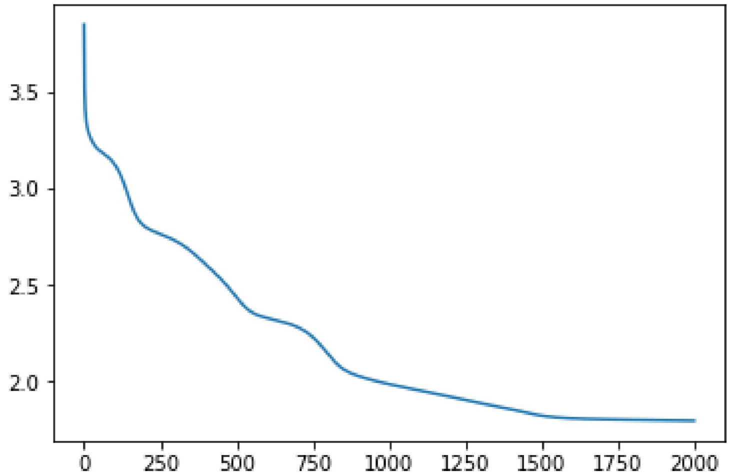
```
10 25
```

In [91]:

```
# NN1 with m = 10
m = 10
w1 = np.random.rand(m,n)*0.1
w2 = np.random.rand(m,m)*0.1
w3 = np.random.rand(n,m)*0.1
loss_list = []
for steps in range(2000):
    Gradient_w1 = np.zeros(w1.shape)
    Gradient_w2 = np.zeros(w2.shape)
    Gradient_w3 = np.zeros(w3.shape)
    total_loss = 0
    for col in range(t):
        loss = NN1_forward_pass(x[:,col],y[:,col],w1,w2,w3)
        total_loss += np.linalg.norm(loss)**2
        dfdw1,dfdw2,dfdw3 = NN1_backprop(x[:,col],w1,w2,w3,n,m)
        Gradient_w1 += VecXTen(loss,dfdw1)
        Gradient_w2 += VecXTen(loss,dfdw2)
        Gradient_w3 += VecXTen(loss,dfdw3)
    Gradient_w1 = Gradient_w1 * 2 / t
    Gradient_w2 = Gradient_w2 * 2 / t
    Gradient_w3 = Gradient_w3 * 2 / t
    w1 = w1 - alpha * Gradient_w1
    w2 = w2 - alpha * Gradient_w2
    w3 = w3 - alpha * Gradient_w3
    #print("at iteration ", steps, " loss is ", total_loss/t)
    loss_list.append(total_loss/t)
print("Final Loss: ", total_loss/t)
#print("final W2 * W1: ")
#print(w2.dot(w1))
pyp.plot(loss_list[1:])
pyp.show()

    #print("GradientW1: ")
    #print(Gradient_w1)
    #print("loss:")
    #print(loss)
    #print("dfdw1:")
    #print(dfdw1)
    #print("dfdw2:")
    #print(dfdw2)
```

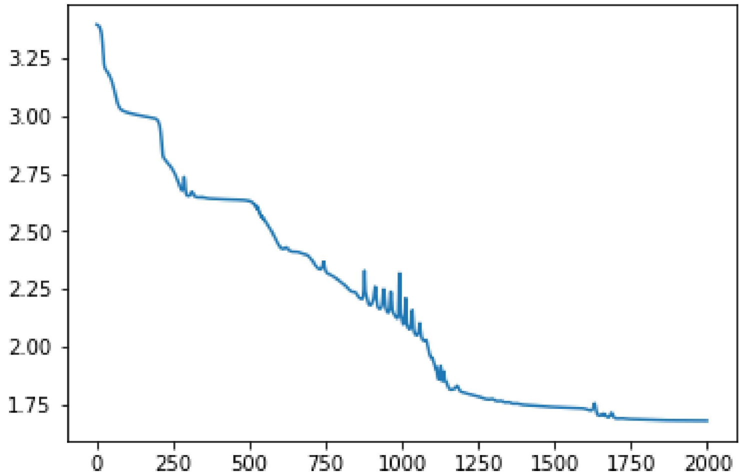
```
Final Loss:  1.7962366226647475
```



In [92]:
```
# NN1 with m = 25
m = 25
alpha = 0.05
w1 = np.random.rand(m,n)*0.01
w2 = np.random.rand(m,m)*0.01
w3 = np.random.rand(n,m)*0.01
loss_list = []
for steps in range(2000):
    Gradient_w1 = np.zeros(w1.shape)
    Gradient_w2 = np.zeros(w2.shape)
    Gradient_w3 = np.zeros(w3.shape)
    total_loss = 0
    for col in range(t):
        loss = NN1_forward_pass(x[:,col],y[:,col],w1,w2,w3)
        total_loss += np.linalg.norm(loss)**2
        dfdw1,dfdw2,dfdw3 = NN1_backprop(x[:,col],w1,w2,w3,n,m)
        Gradient_w1 += VecXTen(loss,dfdw1)
        Gradient_w2 += VecXTen(loss,dfdw2)
        Gradient_w3 += VecXTen(loss,dfdw3)
    Gradient_w1 = Gradient_w1 * 2 / t
    Gradient_w2 = Gradient_w2 * 2 / t
    Gradient_w3 = Gradient_w3 * 2 / t
    w1 = w1 - alpha * Gradient_w1
    w2 = w2 - alpha * Gradient_w2
    w3 = w3 - alpha * Gradient_w3
    #print("at iteration ", steps, " loss is ", total_loss/t)
    loss_list.append(total_loss/t)
print("Final Loss: ", total_loss/t)
#print("final W2 * W1: ")
#print(w2.dot(w1))
pyp.plot(loss_list[1:])
pyp.show()

    #print("GradientW1: ")
    #print(Gradient_w1)
    #print("loss:")
    #print(loss)
    #print("dfdw1:")
    #print(dfdw1)
    #print("dfdw2:")
    #print(dfdw2)
```

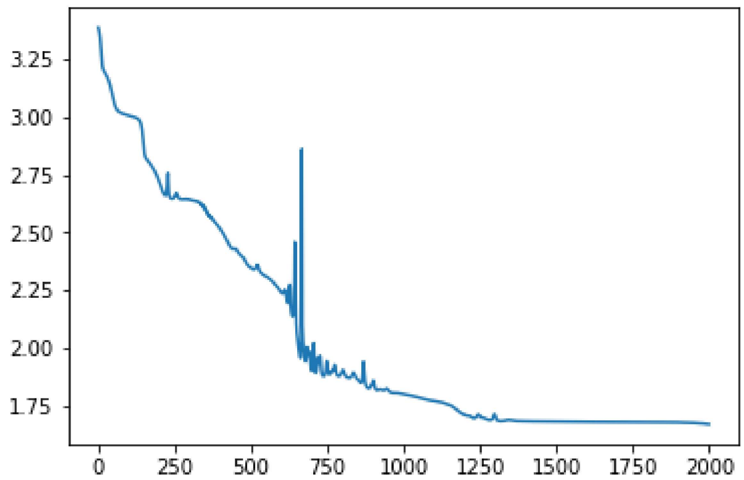
```
Final Loss:  1.6788260271653126
```



In [95]:
```
# NN1 with m = 50
m = 50
alpha = 0.05
w1 = np.random.rand(m,n)*0.01
w2 = np.random.rand(m,m)*0.01
w3 = np.random.rand(n,m)*0.01
loss_list = []
for steps in range(2000):
    Gradient_w1 = np.zeros(w1.shape)
    Gradient_w2 = np.zeros(w2.shape)
    Gradient_w3 = np.zeros(w3.shape)
    total_loss = 0
    for col in range(t):
        loss = NN1_forward_pass(x[:,col],y[:,col],w1,w2,w3)
        total_loss += np.linalg.norm(loss)**2
        dfdw1,dfdw2,dfdw3 = NN1_backprop(x[:,col],w1,w2,w3,n,m)
        Gradient_w1 += VecXTen(loss,dfdw1)
        Gradient_w2 += VecXTen(loss,dfdw2)
        Gradient_w3 += VecXTen(loss,dfdw3)
    Gradient_w1 = Gradient_w1 * 2 / t
    Gradient_w2 = Gradient_w2 * 2 / t
    Gradient_w3 = Gradient_w3 * 2 / t
    w1 = w1 - alpha * Gradient_w1
    w2 = w2 - alpha * Gradient_w2
    w3 = w3 - alpha * Gradient_w3
    #print("at iteration ", steps, " loss is ", total_loss/t)
    loss_list.append(total_loss/t)
print("Final Loss: ", total_loss/t)
#print("final W2 * W1: ")
#print(w2.dot(w1))
pyp.plot(loss_list[1:])
pyp.show()

    #print("GradientW1: ")
    #print(Gradient_w1)
    #print("loss:")
    #print(loss)
    #print("dfdw1:")
    #print(dfdw1)
    #print("dfdw2:")
    #print(dfdw2)
```

```
Final Loss:  1.668421258126552
```



In [119]:
```
x = np.load("assignment8_X.npy")
y = np.load("assignment8_Y.npy")
x = x.T
y = y.T

alpha = 0.01
n = x.shape[0] # n = 10
t = x.shape[1] # t = 25
def vec_sigmod(x):
    for i in range(len(x)):
        x[i] = 1/(1+np.exp(-x[i]))
    return x
def sigmod(x):
    return 1/(1+np.exp(-x))
def Sigmod_forward_pass(x,y,w1,w2,w3):
    #print("before forward pass, x")
    #print(x)
    #print("w2")
    #print(w2)
    #print("w1")
    #print(w1)
    #print("y")
    #print(y)
    fwx = w3.dot(vec_sigmod(w2.dot(vec_sigmod(w1.dot(x)))))
    #print("fwx")
    #print(fwx)
    loss = fwx - y
    #print("loss")
    #print(loss)
    return loss

def Sigmod_backprop(x,w1,w2,w3,n,m): # w1 25x10 w2 10x25
    dfdw3 = np.zeros((n,n,m))
    dfdw2 = np.zeros((n,m,m))
    dfdw1 = np.zeros((n,m,n))

    # Compute dv2dw2, dv1dw1
    dv2dw2 = np.zeros((m,m,m))
    dv1dw1 = np.zeros((m,m,n))
    h1 = vec_sigmod(w1.dot(x))
    h2 = vec_sigmod(w2.dot(h1))
    for i in range(m):
        dv2dw2[i][i] = h1.T
        dv1dw1[i][i] = x.T

    # Compute dfdv2,dfdv1
    dh2dv2 = np.zeros((m,m))    # 25x25
    dh1dv1 = np.zeros((m,m))    # 25x25
    v1 = w1.dot(x) # 25x1
    v2 = w2.dot(vec_sigmod(v1))
    for i in range(m):
        #dh2dv[i][i] = sigmod(v[i])(1-sigmod(v[i]))
        dh2dv2[i][i] = h2[i]*(1-h2[i])
        dh1dv1[i][i] = h1[i]*(1-h1[i])
    dfdv2 = w3.dot(dh2dv2)     # nxm
    dfdv1 = w3.dot(dh2dv2.dot(w2.dot(dh1dv1)))      # nxm
```

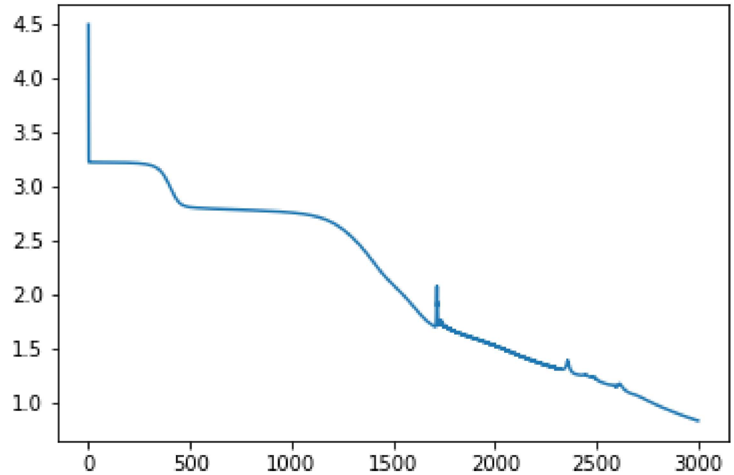
```
    # Compute dfdw1,dfdw2,dfdw3
    for i in range (n):
        for j in range(m):
            #print(dv2dw2.shape,dfdv2.shape)
            dfdw2[i] += dv2dw2[j]*dfdv2[i][j]
            dfdw1[i] += dv1dw1[j]*dfdv1[i][j]
    for i in range(n):
        dfdw3[i][i] = h2.T

    return dfdw1,dfdw2,dfdw3
```

In [80]:
```
# NN2 with m = 10
alpha = 0.1
m = 10
w1 = np.random.rand(m,n)*0.1 # 10x10
w2 = np.random.rand(m,m)*0.1 # 10x10
w3 = np.random.rand(n,m)*0.1 # 10x10

loss_list = []
for steps in range(3000):
    Gradient_w1 = np.zeros(w1.shape)
    Gradient_w2 = np.zeros(w2.shape)
    Gradient_w3 = np.zeros(w3.shape)
    total_loss = 0
    for col in range(t):
        loss = Sigmod_forward_pass(x[:,col],y[:,col],w1,w2,w3)
        total_loss += np.linalg.norm(loss)**2
        dfdw1,dfdw2,dfdw3 = Sigmod_backprop(x[:,col],w1,w2,w3,n,m)
        Gradient_w1 += VecXTen(loss,dfdw1)
        Gradient_w2 += VecXTen(loss,dfdw2)
        Gradient_w3 += VecXTen(loss,dfdw3)
    Gradient_w1 = Gradient_w1 * 2 / t
    Gradient_w2 = Gradient_w2 * 2 / t
    Gradient_w3 = Gradient_w3 * 2 / t
    w1 = w1 - alpha * Gradient_w1
    w2 = w2 - alpha * Gradient_w2
    w3 = w3 - alpha * Gradient_w3
    #print("at iteration ", steps, " loss is ", total_loss/t)
    loss_list.append(total_loss/t)
print("Final Loss: ", total_loss/t)
#print("final W2 * W1: ")
#print(w2.dot(w1))
pyp.plot(loss_list)
pyp.show()
    #print("GradientW1: ")
    #print(Gradient_w1)
    #print("loss:")
    #print(loss)
    #print("dfdw1:")
    #print(dfdw1)
    #print("dfdw2:")
    #print(dfdw2)
```

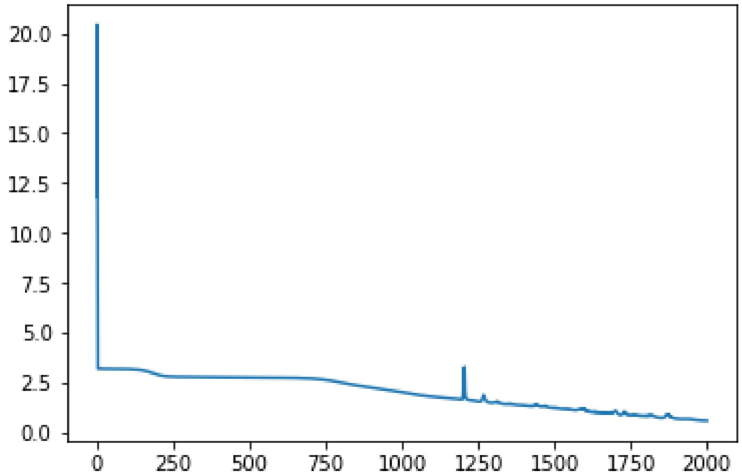
```
Final Loss:  0.8272811067483257
```



In [112]:
```
# NN2 with m = 25
alpha = 0.1
m = 25
w1 = np.random.rand(m,n)*0.1 # 10x10
w2 = np.random.rand(m,m)*0.1 # 10x10
w3 = np.random.rand(n,m)*0.1 # 10x10

loss_list = []
for steps in range(2000):
    Gradient_w1 = np.zeros(w1.shape)
    Gradient_w2 = np.zeros(w2.shape)
    Gradient_w3 = np.zeros(w3.shape)
    total_loss = 0
    for col in range(t):
        loss = Sigmod_forward_pass(x[:,col],y[:,col],w1,w2,w3)
        total_loss += np.linalg.norm(loss)**2
        dfdw1,dfdw2,dfdw3 = Sigmod_backprop(x[:,col],w1,w2,w3,n,m)
        Gradient_w1 += VecXTen(loss,dfdw1)
        Gradient_w2 += VecXTen(loss,dfdw2)
        Gradient_w3 += VecXTen(loss,dfdw3)
    Gradient_w1 = Gradient_w1 * 2 / t
    Gradient_w2 = Gradient_w2 * 2 / t
    Gradient_w3 = Gradient_w3 * 2 / t
    w1 = w1 - alpha * Gradient_w1
    w2 = w2 - alpha * Gradient_w2
    w3 = w3 - alpha * Gradient_w3
    #print("at iteration ", steps, " loss is ", total_loss/t)
    loss_list.append(total_loss/t)
print("Final Loss: ", total_loss/t)
#print("final W2 * W1: ")
#print(w2.dot(w1))
pyp.plot(loss_list)
pyp.show()
    #print("GradientW1: ")
    #print(Gradient_w1)
    #print("loss:")
    #print(loss)
    #print("dfdw1:")
    #print(dfdw1)
    #print("dfdw2:")
    #print(dfdw2)
```

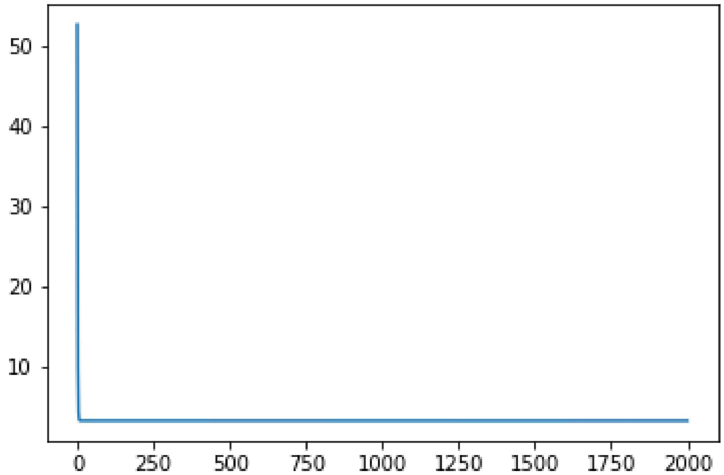
```
Final Loss:  0.6054593372685161
```



In [120]:
```
# NN2 with m = 50
alpha = 0.005
m = 50
w1 = np.random.rand(m,n)*0.1 # 10x10
w2 = np.random.rand(m,m)*0.1 # 10x10
w3 = np.random.rand(n,m)*0.1 # 10x10

loss_list = []
for steps in range(2000):
    Gradient_w1 = np.zeros(w1.shape)
    Gradient_w2 = np.zeros(w2.shape)
    Gradient_w3 = np.zeros(w3.shape)
    total_loss = 0
    for col in range(t):
        loss = Sigmod_forward_pass(x[:,col],y[:,col],w1,w2,w3)
        total_loss += np.linalg.norm(loss)**2
        dfdw1,dfdw2,dfdw3 = Sigmod_backprop(x[:,col],w1,w2,w3,n,m)
        Gradient_w1 += VecXTen(loss,dfdw1)
        Gradient_w2 += VecXTen(loss,dfdw2)
        Gradient_w3 += VecXTen(loss,dfdw3)
    Gradient_w1 = Gradient_w1 * 2 / t
    Gradient_w2 = Gradient_w2 * 2 / t
    Gradient_w3 = Gradient_w3 * 2 / t
    w1 = w1 - alpha * Gradient_w1
    w2 = w2 - alpha * Gradient_w2
    w3 = w3 - alpha * Gradient_w3
    print("at iteration ", steps, " loss is ", total_loss/t,end = '\r')
    loss_list.append(total_loss/t)
print("Final Loss: ", total_loss/t)
#print("final W2 * W1: ")
#print(w2.dot(w1))
pyp.plot(loss_list)
pyp.show()
    #print("GradientW1: ")
    #print(Gradient_w1)
    #print("loss:")
    #print(loss)
    #print("dfdw1:")
    #print(dfdw1)
    #print("dfdw2:")
    #print(dfdw2)
```

```
Final Loss:  3.2164977948406766
```



In [117]:
```
# Linear SVM NN
X = np.load("assignment9_X.npy")
Y = np.load("assignment9_Y.npy")
t = np.shape(X)[1]   # number of sameples 50 (=np.shape(Y))
n = np.shape(X)[0]   # dimension of samples 10
k = 10               # number of classes
L = 0                # loss
total_loss = 0       # total loss
L_list = []
alpha = 0.0012         # learning rate
W = np.random.rand(k,n)*0.01 # weight matrix
print(W)
fix_margin = 1
for steps in range(500):
    dLdW = np.zeros((k,n))    # partial devirative of L with respect to W
    for i in range(t):
        classOfx = Y[i]
        counter = 0
        for j in range(k):
            if j != classOfx:
                L += max(0,W[j].dot(X[:,i])-W[classOfx].dot(X[:,i])+fix_margin
)
                #print(X[:,i].shape)
                if(W[j].dot(X[:,i])-W[classOfx].dot(X[:,i])+fix_margin > 0):
                    counter += 1
                total_loss += L
                L = 0
        dLdW[classOfx] += -1 * counter * X[:,i].T
    W = W - alpha * dLdW
    print("at iteration ", steps, " loss is ", total_loss)
    L_list.append(total_loss)
    total_loss = 0
pyp.plot(L_list)
pyp.show()
```

```
[[8.82170745e-03 6.51767792e-03 4.83170777e-04 3.12632509e-03
  3.44929992e-03 6.32485414e-04 6.70754340e-03 9.67863364e-03
  9.74184241e-03 3.15140658e-03]
 [3.42461545e-03 1.16613653e-03 4.42279269e-03 3.18236657e-03
  2.66230012e-03 5.93047398e-03 8.32181071e-04 9.74430721e-04
  9.16559238e-03 5.19474588e-03]
 [3.41637217e-03 7.65376893e-03 4.89664086e-05 1.84454659e-03
  1.94596193e-03 6.51725776e-03 9.95429781e-03 3.39671833e-03
  9.74560035e-03 1.38977420e-03]
 [3.79535977e-03 4.55256780e-03 8.51263717e-03 2.91048980e-03
  2.67674836e-03 1.44121323e-03 7.43007463e-03 6.45583271e-03
  7.45040314e-03 7.36814178e-03]
 [2.03593260e-04 4.66136968e-03 5.07788448e-04 8.60040055e-03
  4.69964345e-03 3.98514680e-04 4.19040449e-03 4.57151144e-03
  6.03586983e-03 9.44339595e-03]
 [7.39933741e-03 6.16435030e-03 1.58942011e-03 9.83257620e-03
  8.60852475e-03 4.56539998e-03 7.17912102e-03 5.49473489e-03
  4.37423798e-03 9.48407957e-03]
 [1.27854065e-03 2.10066329e-03 6.95080975e-04 4.25437902e-03
  1.17729550e-03 6.38903193e-03 4.19433956e-03 7.52156802e-03
  8.43708843e-04 9.57001501e-03]
 [6.87275484e-03 7.04523792e-03 9.17211902e-03 8.62264233e-03
  8.14242901e-03 9.91243063e-03 5.20342050e-03 6.70041750e-03
  9.60506756e-03 9.93533579e-03]
 [5.40046408e-03 6.43865317e-03 3.06960023e-03 7.54368293e-03
  1.88765042e-03 4.25707131e-03 6.00494248e-03 3.12734824e-03
  2.19792357e-03 4.80650666e-03]
 [8.65927637e-03 1.43687575e-03 2.06771422e-03 4.73219682e-03
  6.62023161e-03 5.46751672e-03 3.51694815e-03 7.78769119e-03
  6.61765436e-03 5.87835841e-03]]
at iteration  0  loss is  445.6325758191826
at iteration  1  loss is  75.40443541049389
at iteration  2  loss is  28.694629213406166
at iteration  3  loss is  16.870884295660918
at iteration  4  loss is  13.45358840597196
at iteration  5  loss is  11.16285709660814
at iteration  6  loss is  9.989157528315022
at iteration  7  loss is  8.985071461519949
at iteration  8  loss is  8.013844581192785
at iteration  9  loss is  7.290567039195828
at iteration  10  loss is  7.110961238177746
at iteration  11  loss is  6.756232540383241
at iteration  12  loss is  6.412565593667965
at iteration  13  loss is  6.032772680275565
at iteration  14  loss is  5.912664967727434
at iteration  15  loss is  5.7362535461697135
at iteration  16  loss is  5.859867446697669
at iteration  17  loss is  5.491841490601571
at iteration  18  loss is  5.427853970831974
at iteration  19  loss is  5.4362167983950105
at iteration  20  loss is  5.362323016006433
at iteration  21  loss is  5.200597773853459
at iteration  22  loss is  5.143301615139715
at iteration  23  loss is  5.036757491387791
at iteration  24  loss is  4.974202679983251
at iteration  25  loss is  4.776677073022083
at iteration  26  loss is  4.729100689439595
```

```
at iteration  27  loss is  4.683093375767494
at iteration  28  loss is  4.5289347120920915
at iteration  29  loss is  4.524212569432346
at iteration  30  loss is  4.543268112339181
at iteration  31  loss is  4.359712767149455
at iteration  32  loss is  4.3345638625673155
at iteration  33  loss is  4.175104295418633
at iteration  34  loss is  4.096370006504833
at iteration  35  loss is  4.021126931053947
at iteration  36  loss is  3.927364173598807
at iteration  37  loss is  3.7884984064651963
at iteration  38  loss is  3.762209792259607
at iteration  39  loss is  3.612237171526668
at iteration  40  loss is  3.485623375292114
at iteration  41  loss is  3.514916208941019
at iteration  42  loss is  3.3273459532736185
at iteration  43  loss is  3.3217607556358764
at iteration  44  loss is  3.09652397711326
at iteration  45  loss is  3.0337147340207755
at iteration  46  loss is  3.0375763176502524
at iteration  47  loss is  2.8228962515757914
at iteration  48  loss is  2.792045172060181
at iteration  49  loss is  2.6979461669889986
at iteration  50  loss is  2.5024453385218663
at iteration  51  loss is  2.364886559817924
at iteration  52  loss is  2.219782827097463
at iteration  53  loss is  2.1875332224931627
at iteration  54  loss is  2.106644393273104
at iteration  55  loss is  2.046396383321575
at iteration  56  loss is  1.9544765588033766
at iteration  57  loss is  1.8308786082316661
at iteration  58  loss is  1.7351765282874219
at iteration  59  loss is  1.6374143425825052
at iteration  60  loss is  1.5589796821312198
at iteration  61  loss is  1.5474343541816555
at iteration  62  loss is  1.47564957449121
at iteration  63  loss is  1.3567598206395601
at iteration  64  loss is  1.2745806223657956
at iteration  65  loss is  1.2576952849131642
at iteration  66  loss is  1.1817159569805264
at iteration  67  loss is  1.0203604577781311
at iteration  68  loss is  0.9453778874024508
at iteration  69  loss is  0.9020608313286438
at iteration  70  loss is  0.84717493780194
at iteration  71  loss is  0.855315504857618
at iteration  72  loss is  0.7712685105203256
at iteration  73  loss is  0.6521694833856211
at iteration  74  loss is  0.607437105645702
at iteration  75  loss is  0.5541627798205955
at iteration  76  loss is  0.5824761152474074
at iteration  77  loss is  0.5295831241348958
at iteration  78  loss is  0.4212252772068741
at iteration  79  loss is  0.37286683879849036
at iteration  80  loss is  0.2982537100866791
at iteration  81  loss is  0.26438153778796636
at iteration  82  loss is  0.21146318353641114
at iteration  83  loss is  0.1319344111036802
```

```
at iteration  84  loss is  0.09634831369308117
at iteration  85  loss is  0.11776651664328952
at iteration  86  loss is  0.07021323093616516
at iteration  87  loss is  0.08420905867520379
at iteration  88  loss is  0.0593744295858496
at iteration  89  loss is  0.059361736421451816
at iteration  90  loss is  0.0976269942789969
at iteration  91  loss is  0.046043333841330636
at iteration  92  loss is  0.06049941141682247
at iteration  93  loss is  0.03498961854172755
at iteration  94  loss is  0.03579940992414965
at iteration  95  loss is  0.06475577364478902
at iteration  96  loss is  0.023578229722454047
at iteration  97  loss is  0.04998606806184558
at iteration  98  loss is  0.01902253025094769
at iteration  99  loss is  0.023038996311553817
at iteration  100  loss is  0.038244894047931854
at iteration  101  loss is  0.013859158468759958
at iteration  102  loss is  0.0239563463817678
at iteration  103  loss is  0.045750774828656304
at iteration  104  loss is  0.03366077593338712
at iteration  105  loss is  0
at iteration  106  loss is  0
at iteration  107  loss is  0
at iteration  108  loss is  0
at iteration  109  loss is  0
at iteration  110  loss is  0
at iteration  111  loss is  0
at iteration  112  loss is  0
at iteration  113  loss is  0
at iteration  114  loss is  0
at iteration  115  loss is  0
at iteration  116  loss is  0
at iteration  117  loss is  0
at iteration  118  loss is  0
at iteration  119  loss is  0
at iteration  120  loss is  0
at iteration  121  loss is  0
at iteration  122  loss is  0
at iteration  123  loss is  0
at iteration  124  loss is  0
at iteration  125  loss is  0
at iteration  126  loss is  0
at iteration  127  loss is  0
at iteration  128  loss is  0
at iteration  129  loss is  0
at iteration  130  loss is  0
at iteration  131  loss is  0
at iteration  132  loss is  0
at iteration  133  loss is  0
at iteration  134  loss is  0
at iteration  135  loss is  0
at iteration  136  loss is  0
at iteration  137  loss is  0
at iteration  138  loss is  0
at iteration  139  loss is  0
at iteration  140  loss is  0
```

```
at iteration  141  loss is  0
at iteration  142  loss is  0
at iteration  143  loss is  0
at iteration  144  loss is  0
at iteration  145  loss is  0
at iteration  146  loss is  0
at iteration  147  loss is  0
at iteration  148  loss is  0
at iteration  149  loss is  0
at iteration  150  loss is  0
at iteration  151  loss is  0
at iteration  152  loss is  0
at iteration  153  loss is  0
at iteration  154  loss is  0
at iteration  155  loss is  0
at iteration  156  loss is  0
at iteration  157  loss is  0
at iteration  158  loss is  0
at iteration  159  loss is  0
at iteration  160  loss is  0
at iteration  161  loss is  0
at iteration  162  loss is  0
at iteration  163  loss is  0
at iteration  164  loss is  0
at iteration  165  loss is  0
at iteration  166  loss is  0
at iteration  167  loss is  0
at iteration  168  loss is  0
at iteration  169  loss is  0
at iteration  170  loss is  0
at iteration  171  loss is  0
at iteration  172  loss is  0
at iteration  173  loss is  0
at iteration  174  loss is  0
at iteration  175  loss is  0
at iteration  176  loss is  0
at iteration  177  loss is  0
at iteration  178  loss is  0
at iteration  179  loss is  0
at iteration  180  loss is  0
at iteration  181  loss is  0
at iteration  182  loss is  0
at iteration  183  loss is  0
at iteration  184  loss is  0
at iteration  185  loss is  0
at iteration  186  loss is  0
at iteration  187  loss is  0
at iteration  188  loss is  0
at iteration  189  loss is  0
at iteration  190  loss is  0
at iteration  191  loss is  0
at iteration  192  loss is  0
at iteration  193  loss is  0
at iteration  194  loss is  0
at iteration  195  loss is  0
at iteration  196  loss is  0
at iteration  197  loss is  0
```

```
at iteration  198  loss is  0
at iteration  199  loss is  0
at iteration  200  loss is  0
at iteration  201  loss is  0
at iteration  202  loss is  0
at iteration  203  loss is  0
at iteration  204  loss is  0
at iteration  205  loss is  0
at iteration  206  loss is  0
at iteration  207  loss is  0
at iteration  208  loss is  0
at iteration  209  loss is  0
at iteration  210  loss is  0
at iteration  211  loss is  0
at iteration  212  loss is  0
at iteration  213  loss is  0
at iteration  214  loss is  0
at iteration  215  loss is  0
at iteration  216  loss is  0
at iteration  217  loss is  0
at iteration  218  loss is  0
at iteration  219  loss is  0
at iteration  220  loss is  0
at iteration  221  loss is  0
at iteration  222  loss is  0
at iteration  223  loss is  0
at iteration  224  loss is  0
at iteration  225  loss is  0
at iteration  226  loss is  0
at iteration  227  loss is  0
at iteration  228  loss is  0
at iteration  229  loss is  0
at iteration  230  loss is  0
at iteration  231  loss is  0
at iteration  232  loss is  0
at iteration  233  loss is  0
at iteration  234  loss is  0
at iteration  235  loss is  0
at iteration  236  loss is  0
at iteration  237  loss is  0
at iteration  238  loss is  0
at iteration  239  loss is  0
at iteration  240  loss is  0
at iteration  241  loss is  0
at iteration  242  loss is  0
at iteration  243  loss is  0
at iteration  244  loss is  0
at iteration  245  loss is  0
at iteration  246  loss is  0
at iteration  247  loss is  0
at iteration  248  loss is  0
at iteration  249  loss is  0
at iteration  250  loss is  0
at iteration  251  loss is  0
at iteration  252  loss is  0
at iteration  253  loss is  0
at iteration  254  loss is  0
```

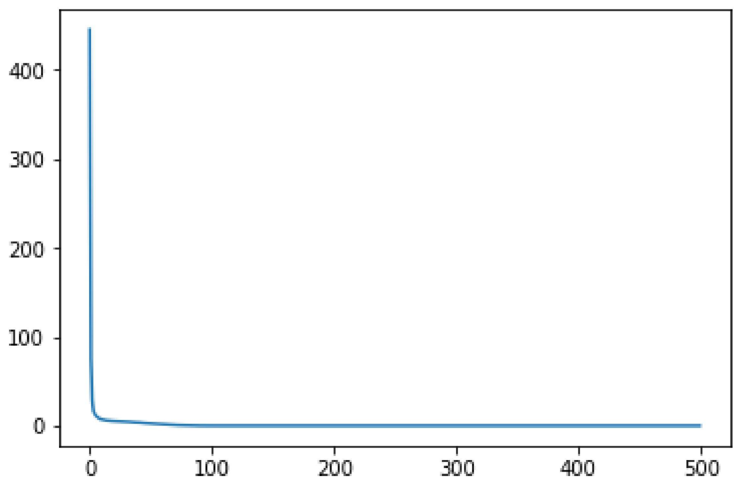
```
at iteration  255  loss is  0
at iteration  256  loss is  0
at iteration  257  loss is  0
at iteration  258  loss is  0
at iteration  259  loss is  0
at iteration  260  loss is  0
at iteration  261  loss is  0
at iteration  262  loss is  0
at iteration  263  loss is  0
at iteration  264  loss is  0
at iteration  265  loss is  0
at iteration  266  loss is  0
at iteration  267  loss is  0
at iteration  268  loss is  0
at iteration  269  loss is  0
at iteration  270  loss is  0
at iteration  271  loss is  0
at iteration  272  loss is  0
at iteration  273  loss is  0
at iteration  274  loss is  0
at iteration  275  loss is  0
at iteration  276  loss is  0
at iteration  277  loss is  0
at iteration  278  loss is  0
at iteration  279  loss is  0
at iteration  280  loss is  0
at iteration  281  loss is  0
at iteration  282  loss is  0
at iteration  283  loss is  0
at iteration  284  loss is  0
at iteration  285  loss is  0
at iteration  286  loss is  0
at iteration  287  loss is  0
at iteration  288  loss is  0
at iteration  289  loss is  0
at iteration  290  loss is  0
at iteration  291  loss is  0
at iteration  292  loss is  0
at iteration  293  loss is  0
at iteration  294  loss is  0
at iteration  295  loss is  0
at iteration  296  loss is  0
at iteration  297  loss is  0
at iteration  298  loss is  0
at iteration  299  loss is  0
at iteration  300  loss is  0
at iteration  301  loss is  0
at iteration  302  loss is  0
at iteration  303  loss is  0
at iteration  304  loss is  0
at iteration  305  loss is  0
at iteration  306  loss is  0
at iteration  307  loss is  0
at iteration  308  loss is  0
at iteration  309  loss is  0
at iteration  310  loss is  0
at iteration  311  loss is  0
```

```
at iteration  312  loss is  0
at iteration  313  loss is  0
at iteration  314  loss is  0
at iteration  315  loss is  0
at iteration  316  loss is  0
at iteration  317  loss is  0
at iteration  318  loss is  0
at iteration  319  loss is  0
at iteration  320  loss is  0
at iteration  321  loss is  0
at iteration  322  loss is  0
at iteration  323  loss is  0
at iteration  324  loss is  0
at iteration  325  loss is  0
at iteration  326  loss is  0
at iteration  327  loss is  0
at iteration  328  loss is  0
at iteration  329  loss is  0
at iteration  330  loss is  0
at iteration  331  loss is  0
at iteration  332  loss is  0
at iteration  333  loss is  0
at iteration  334  loss is  0
at iteration  335  loss is  0
at iteration  336  loss is  0
at iteration  337  loss is  0
at iteration  338  loss is  0
at iteration  339  loss is  0
at iteration  340  loss is  0
at iteration  341  loss is  0
at iteration  342  loss is  0
at iteration  343  loss is  0
at iteration  344  loss is  0
at iteration  345  loss is  0
at iteration  346  loss is  0
at iteration  347  loss is  0
at iteration  348  loss is  0
at iteration  349  loss is  0
at iteration  350  loss is  0
at iteration  351  loss is  0
at iteration  352  loss is  0
at iteration  353  loss is  0
at iteration  354  loss is  0
at iteration  355  loss is  0
at iteration  356  loss is  0
at iteration  357  loss is  0
at iteration  358  loss is  0
at iteration  359  loss is  0
at iteration  360  loss is  0
at iteration  361  loss is  0
at iteration  362  loss is  0
at iteration  363  loss is  0
at iteration  364  loss is  0
at iteration  365  loss is  0
at iteration  366  loss is  0
at iteration  367  loss is  0
at iteration  368  loss is  0
```

```
at iteration  369  loss is  0
at iteration  370  loss is  0
at iteration  371  loss is  0
at iteration  372  loss is  0
at iteration  373  loss is  0
at iteration  374  loss is  0
at iteration  375  loss is  0
at iteration  376  loss is  0
at iteration  377  loss is  0
at iteration  378  loss is  0
at iteration  379  loss is  0
at iteration  380  loss is  0
at iteration  381  loss is  0
at iteration  382  loss is  0
at iteration  383  loss is  0
at iteration  384  loss is  0
at iteration  385  loss is  0
at iteration  386  loss is  0
at iteration  387  loss is  0
at iteration  388  loss is  0
at iteration  389  loss is  0
at iteration  390  loss is  0
at iteration  391  loss is  0
at iteration  392  loss is  0
at iteration  393  loss is  0
at iteration  394  loss is  0
at iteration  395  loss is  0
at iteration  396  loss is  0
at iteration  397  loss is  0
at iteration  398  loss is  0
at iteration  399  loss is  0
at iteration  400  loss is  0
at iteration  401  loss is  0
at iteration  402  loss is  0
at iteration  403  loss is  0
at iteration  404  loss is  0
at iteration  405  loss is  0
at iteration  406  loss is  0
at iteration  407  loss is  0
at iteration  408  loss is  0
at iteration  409  loss is  0
at iteration  410  loss is  0
at iteration  411  loss is  0
at iteration  412  loss is  0
at iteration  413  loss is  0
at iteration  414  loss is  0
at iteration  415  loss is  0
at iteration  416  loss is  0
at iteration  417  loss is  0
at iteration  418  loss is  0
at iteration  419  loss is  0
at iteration  420  loss is  0
at iteration  421  loss is  0
at iteration  422  loss is  0
at iteration  423  loss is  0
at iteration  424  loss is  0
at iteration  425  loss is  0
```

```
at iteration  426  loss is  0
at iteration  427  loss is  0
at iteration  428  loss is  0
at iteration  429  loss is  0
at iteration  430  loss is  0
at iteration  431  loss is  0
at iteration  432  loss is  0
at iteration  433  loss is  0
at iteration  434  loss is  0
at iteration  435  loss is  0
at iteration  436  loss is  0
at iteration  437  loss is  0
at iteration  438  loss is  0
at iteration  439  loss is  0
at iteration  440  loss is  0
at iteration  441  loss is  0
at iteration  442  loss is  0
at iteration  443  loss is  0
at iteration  444  loss is  0
at iteration  445  loss is  0
at iteration  446  loss is  0
at iteration  447  loss is  0
at iteration  448  loss is  0
at iteration  449  loss is  0
at iteration  450  loss is  0
at iteration  451  loss is  0
at iteration  452  loss is  0
at iteration  453  loss is  0
at iteration  454  loss is  0
at iteration  455  loss is  0
at iteration  456  loss is  0
at iteration  457  loss is  0
at iteration  458  loss is  0
at iteration  459  loss is  0
at iteration  460  loss is  0
at iteration  461  loss is  0
at iteration  462  loss is  0
at iteration  463  loss is  0
at iteration  464  loss is  0
at iteration  465  loss is  0
at iteration  466  loss is  0
at iteration  467  loss is  0
at iteration  468  loss is  0
at iteration  469  loss is  0
at iteration  470  loss is  0
at iteration  471  loss is  0
at iteration  472  loss is  0
at iteration  473  loss is  0
at iteration  474  loss is  0
at iteration  475  loss is  0
at iteration  476  loss is  0
at iteration  477  loss is  0
at iteration  478  loss is  0
at iteration  479  loss is  0
at iteration  480  loss is  0
at iteration  481  loss is  0
at iteration  482  loss is  0
```

```
at iteration  483  loss is  0
at iteration  484  loss is  0
at iteration  485  loss is  0
at iteration  486  loss is  0
at iteration  487  loss is  0
at iteration  488  loss is  0
at iteration  489  loss is  0
at iteration  490  loss is  0
at iteration  491  loss is  0
at iteration  492  loss is  0
at iteration  493  loss is  0
at iteration  494  loss is  0
at iteration  495  loss is  0
at iteration  496  loss is  0
at iteration  497  loss is  0
at iteration  498  loss is  0
at iteration  499  loss is  0
```



In [32]:
```
Y
```

Out[32]:
```
array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4,
       4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8,
       8, 9, 9, 9, 9, 9])
```