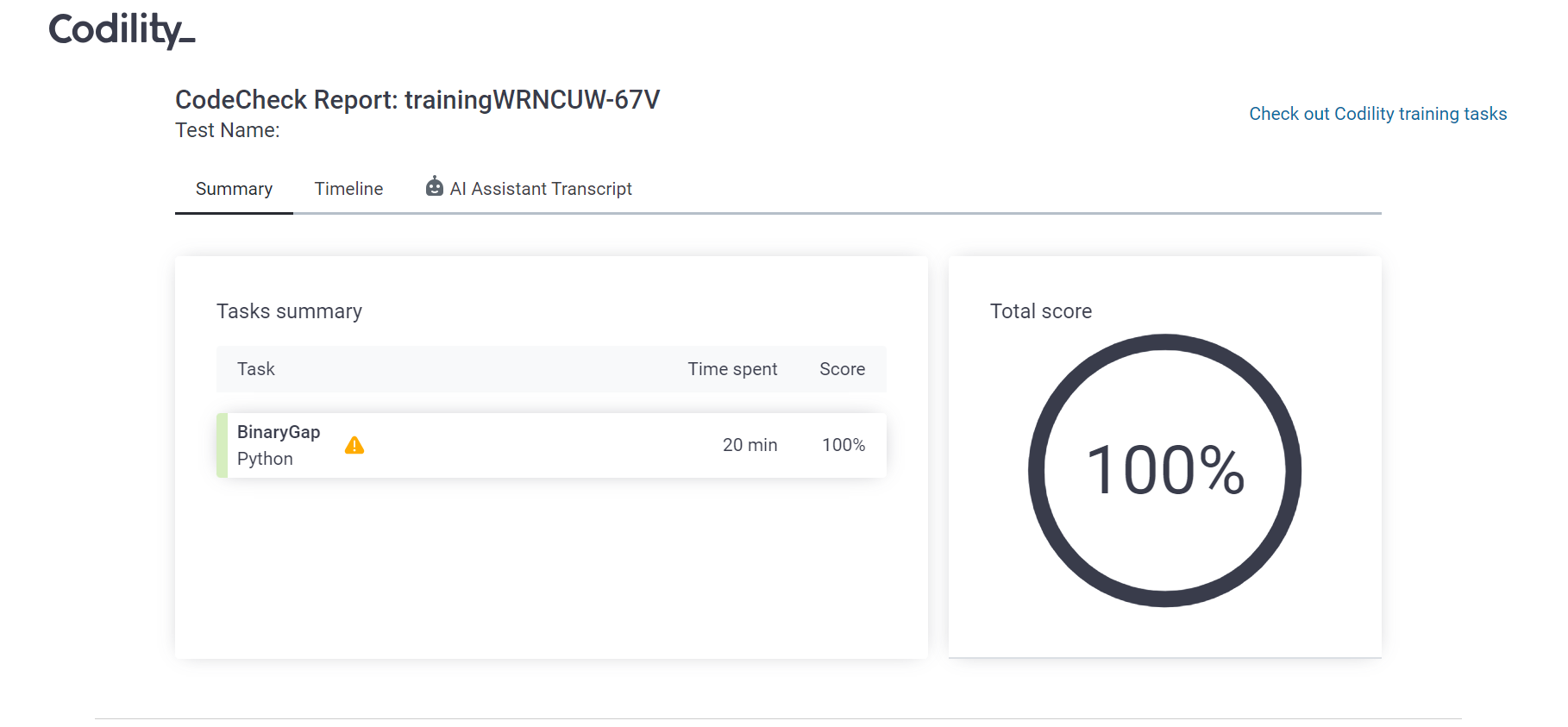
Codility Problems

lesson 1 , Binary gap - <https://app.codility.com/demo/results/trainingWRNCUW-67V/>



def solution(N):

# Implement your solution here

binVal=""

rem=0

while(N>0):

rem=N%2

binVal+=str(rem)

N//=2

print(binVal)

maxlength=0

# count=0

# for i,j in enumerate(binVal):

# if j=="0":

# if i-1>=0 and binVal[i-1]=="1":

# count+=1

# else:

# if count!=0:

# maxlength=max(maxlength,count)

# count=0

# return maxlength

for i in range(len(binVal)):

if binVal[i]=="1" and i+1<len(binVal) and binVal[i+1]=="0":

for j in range(i+1,len(binVal)):

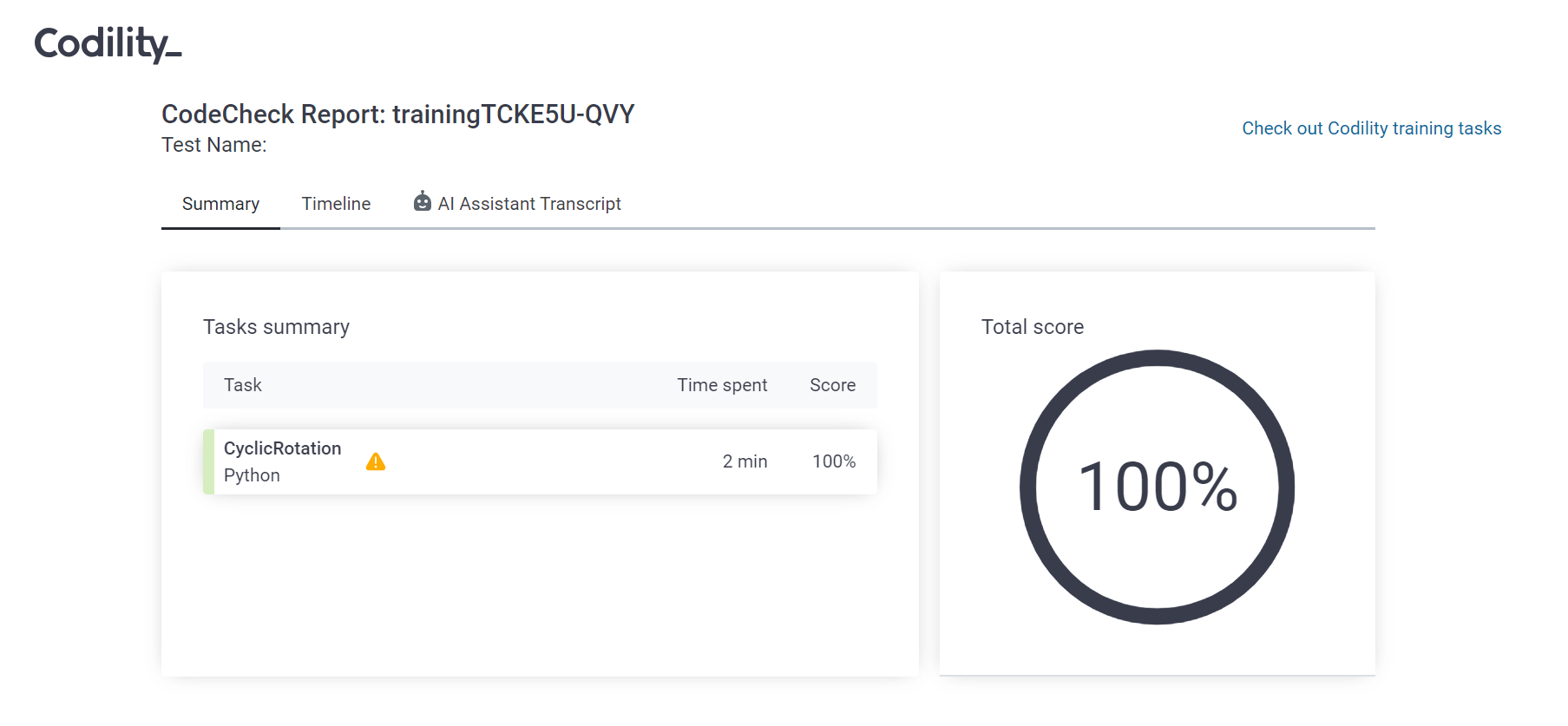
if binVal[j]=="1":

maxlength=max(maxlength,j-i-1)

break

return maxlength

lesson 2 , cyclic rotate - <https://app.codility.com/demo/results/trainingTCKE5U-QVY/>



def solution(A, K):

# Implement your solution here

if A==[]:

return A

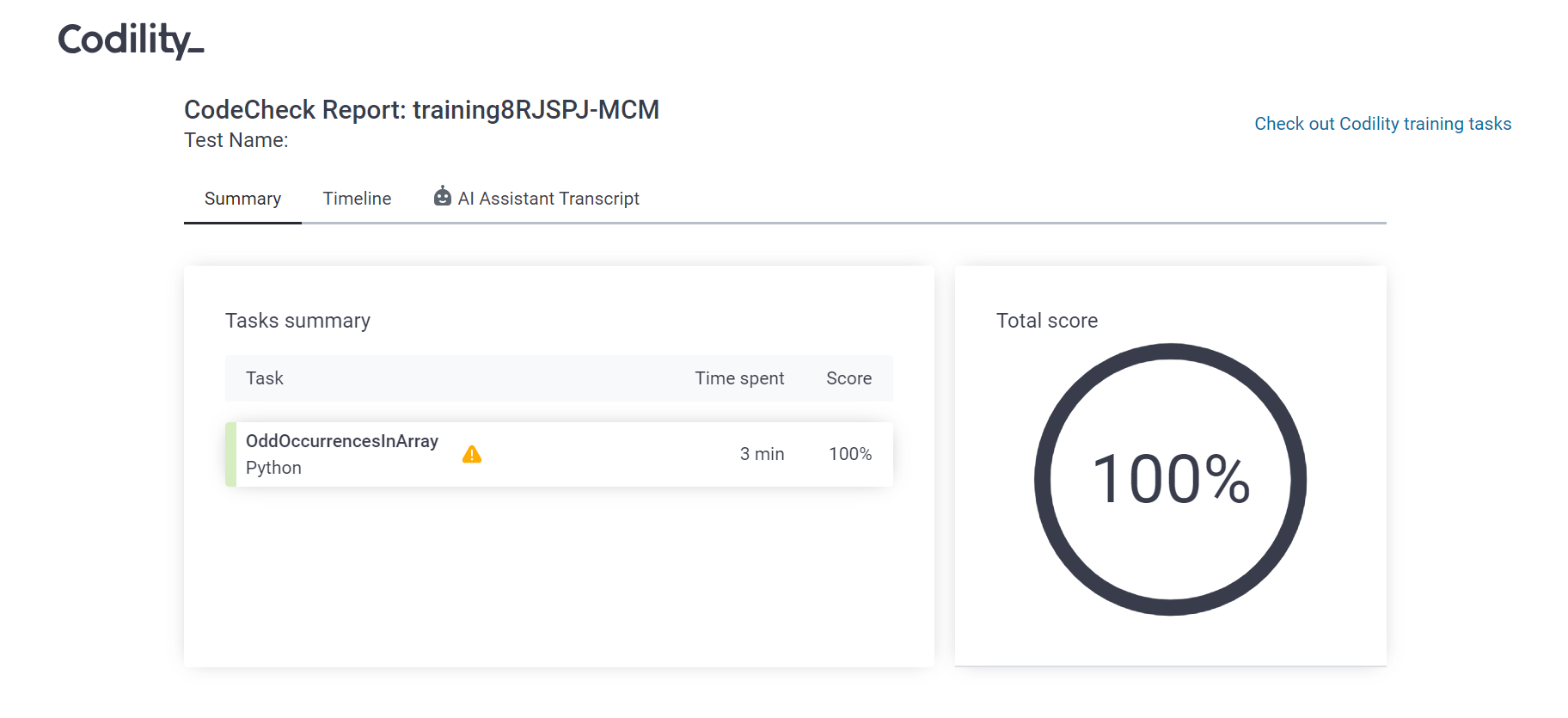
for i in range(K):

temp=A.pop()

A.insert(0,temp)

return A

lesson 2 , odd occurance - <https://app.codility.com/demo/results/training8RJSPJ-MCM/>



def solution(A):

# Implement your solution here

d={}

for i in A:

if i in d:

d[i]+=1

else:

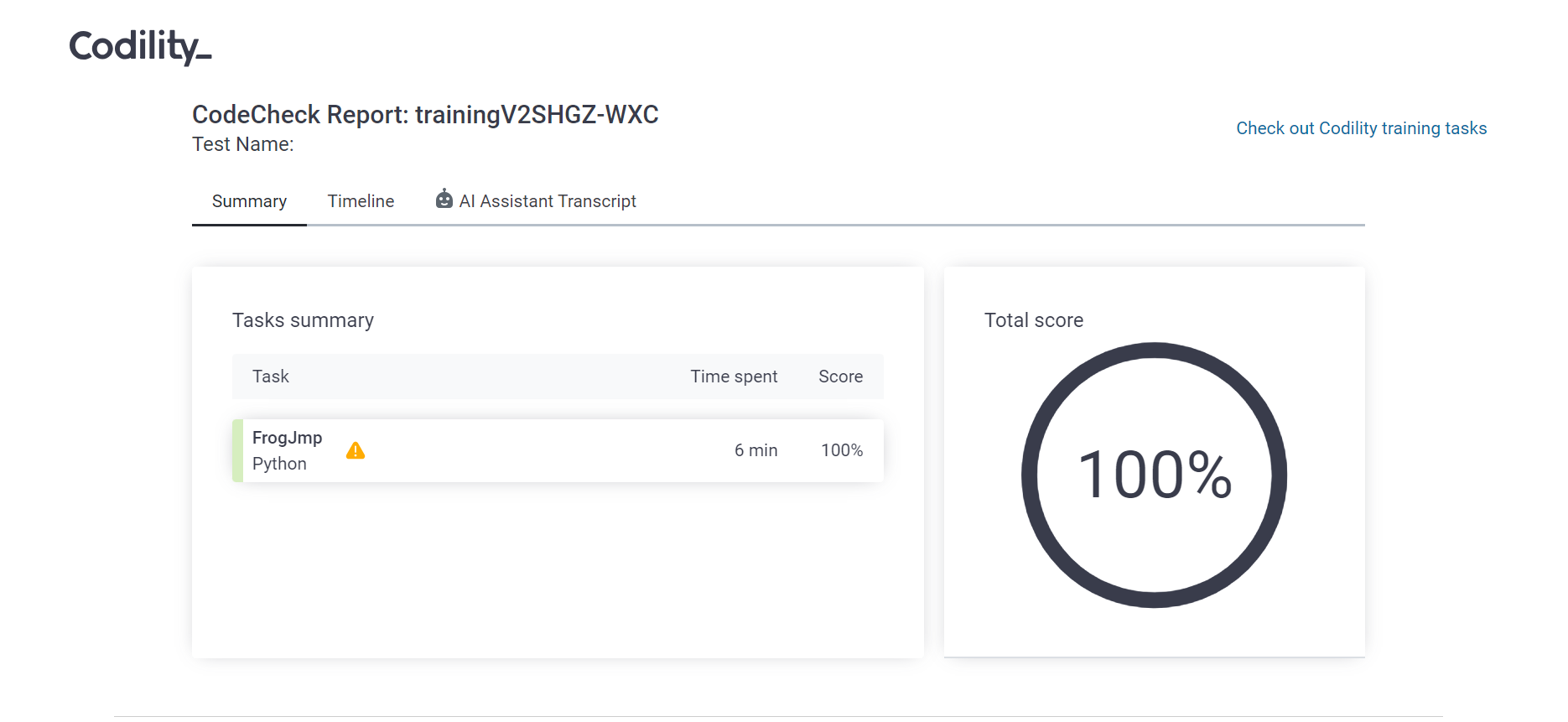
d[i]=1

for i in d:

if d[i]%2!=0:

return i

lesson 3 , frogjump <https://app.codility.com/demo/results/trainingV2SHGZ-WXC/>



def solution(X, Y, D):

# Implement your solution here

temp = (Y-X)//D

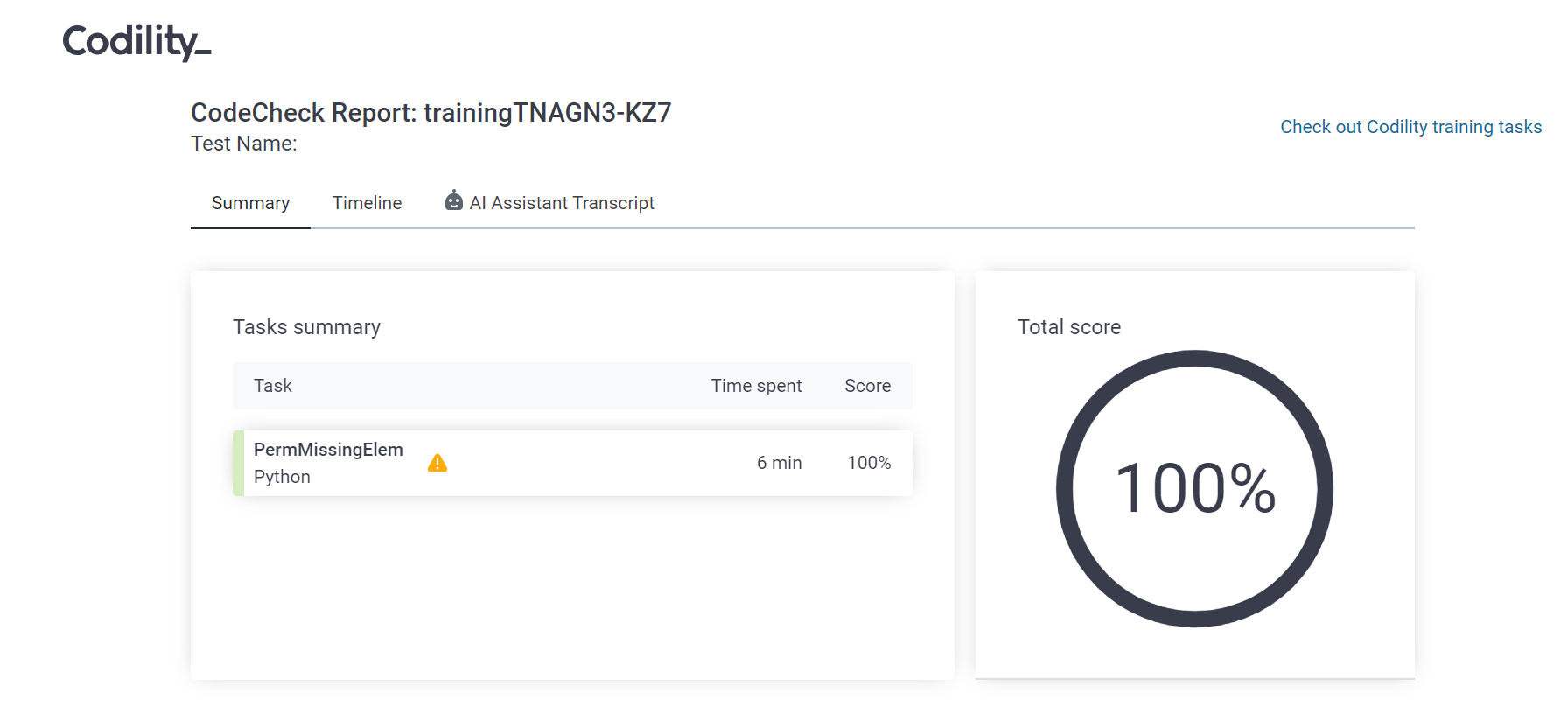
if (X+ (temp\*D) < Y):

return temp+1

else:

return temp

lesson 3, permmissngno - <https://app.codility.com/demo/results/trainingTNAGN3-KZ7/>



def solution(A):

# Implement your solution here

A=sorted(A)

for i in range(len(A)):

if A[i]==i+1:

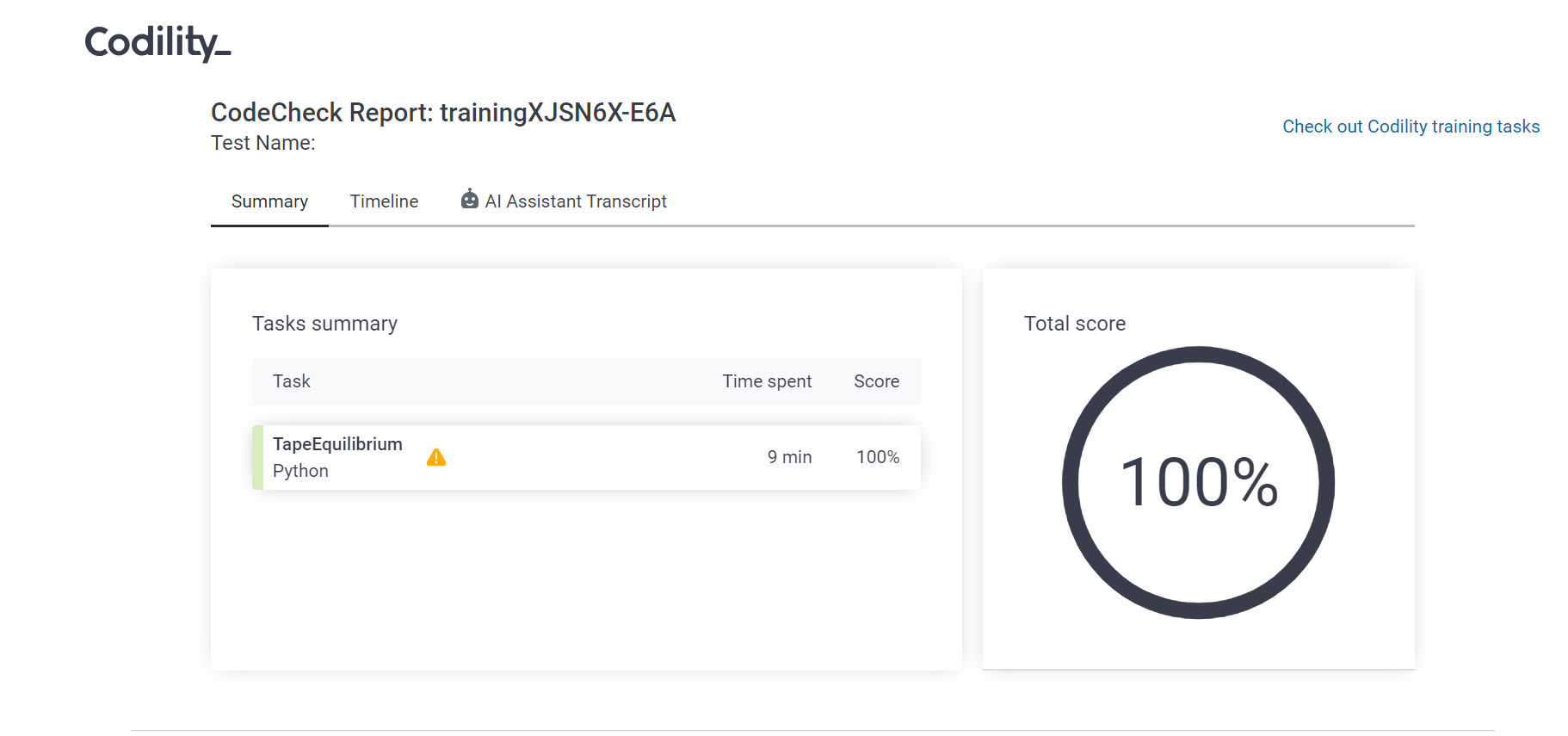
continue

else:

return i+1

return len(A)+1

lesson 3, tapeequilibrium - <https://app.codility.com/demo/results/trainingXJSN6X-E6A/>



def solution(A):

# Implement your solution here

total=sum(A)

minVal= float('inf')

s=0

for i in range(len(A)-1):

s+=A[i]

total-=A[i]

if abs(total - s) < minVal:

minVal=abs(total-s)

# print(total)

# print("minVal"+str(minVal))

return minVal