**Laptop Battery Life**

import sys

data = float(sys.stdin.readline())

training\_data = {}

data\_num = 100

def clean\_input(string):

new\_pts= string.split(",")

return [float(x) for x in new\_pts]

with open("trainingdata.txt","r") as training:

for i in range(0,data\_num):

instance = clean\_input(training.readline())

training\_data[instance[0]] = instance[1]

max\_life = max(training\_data.values())

max\_case = []

lin\_case = {}

for key in training\_data:

if training\_data[key] < max\_life:

lin\_case[key] = training\_data[key]

else:

max\_case.append(key)

full\_battery = min(max\_case)

def sum\_xys(dict):

total = 0

for x in dict:

total += (x \* dict[x])

return float(total

def sum\_xs(dict):

total = 0

for x in dict:

total += x

return float(total)

def sum\_ys(dict):

total = 0

for x in dict:

total += dict[x]

return float(total)

def sum\_x\_sqrd(dict):

total = 0

for x in dict:

total += x\*\*2

return float(total)

def lin\_regr(dict):

n = len(dict)

xys = sum\_xys(dict)

xs = sum\_xs(dict)

ys = sum\_ys(dict)

x\_sqrd = sum\_x\_sqrd(dict)

m = ((n \* xys) - (xs \* ys)) / ((n \* x\_sqrd) - (xs\*\*2))

b = ((x\_sqrd \* ys) - (xs \* xys)) / ((n \* x\_sqrd) - (xs\*\*2))

return [m,b]

lin\_vars = lin\_regr(lin\_case)

def est\_life(chrg\_time):

if chrg\_time >= full\_battery:

batt\_life = max\_life

else:

m = lin\_vars[0]

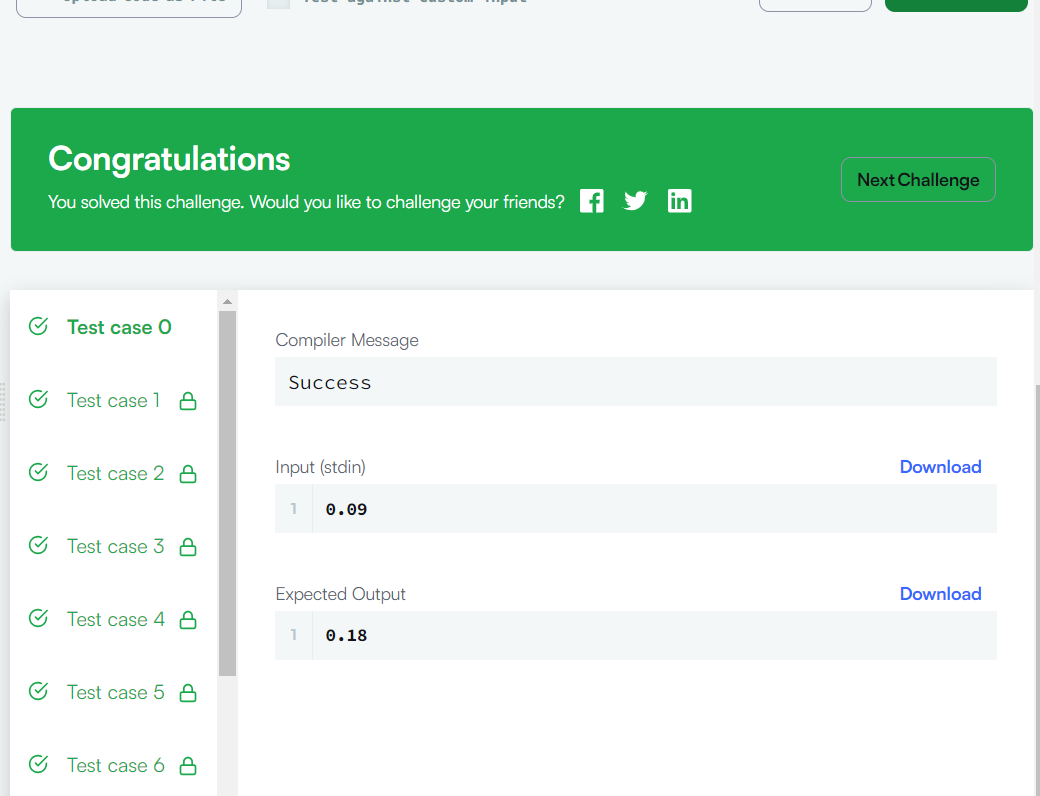
b = lin\_vars[1]

batt\_life = (m \* chrg\_time) + b

return batt\_life

print(est\_life(data))

**Output**

****