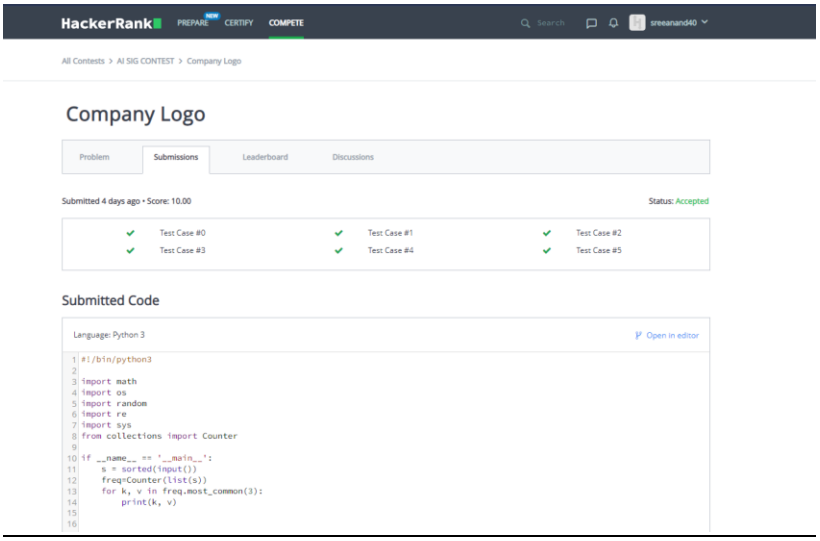
ACM - AI SIG Tasks (Category – 2)

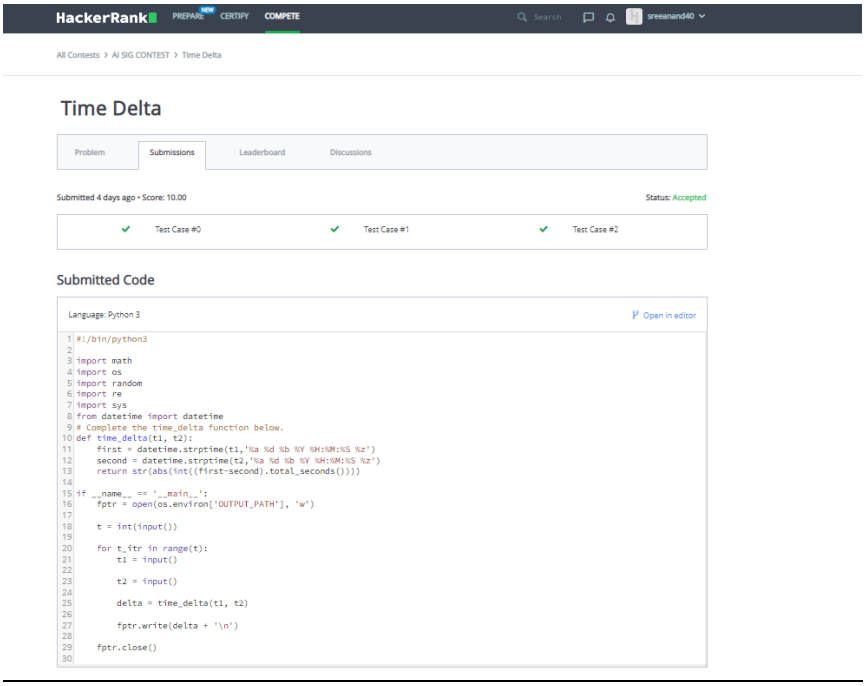
Name: S Anand

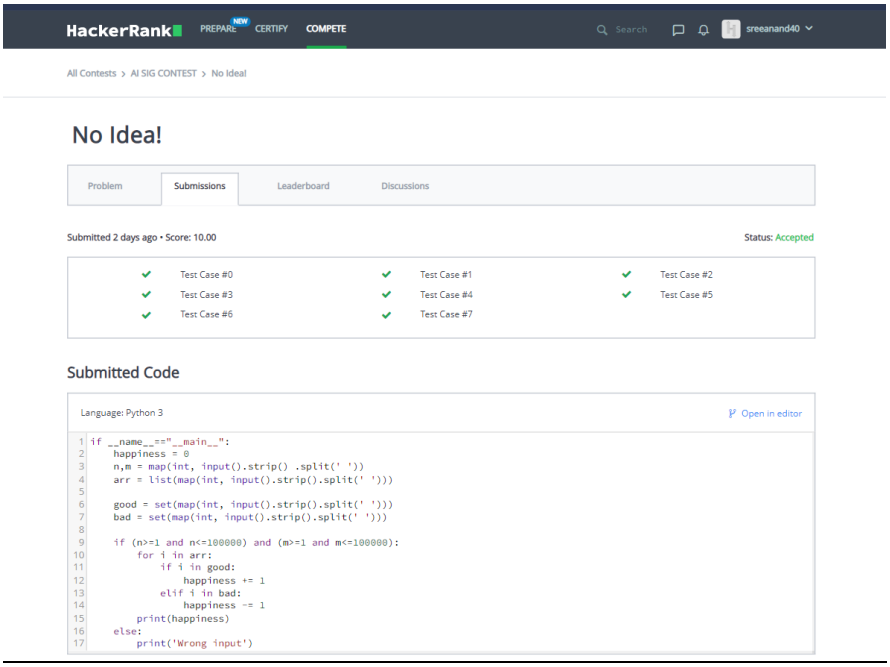
Roll no: AM.EN.U4AIE21155

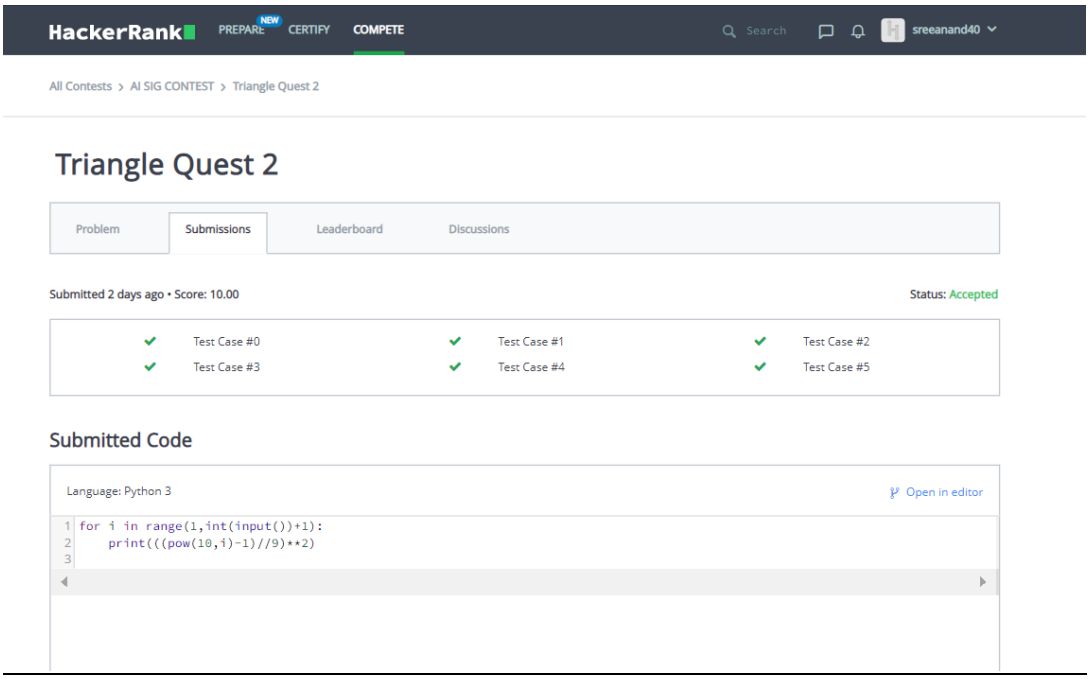
S3 AIE [B]

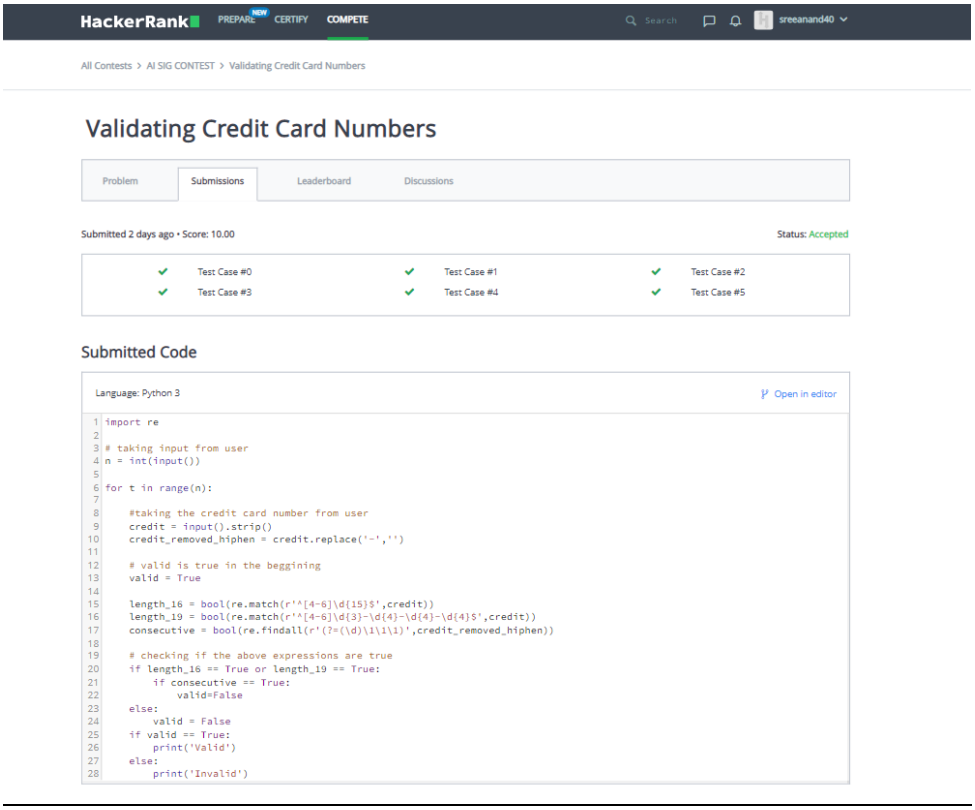
Task – 2






HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
sreeanand40
All Contests › AI SIG CONTEST › No Idea!
No Idea!
Problem
Submissions
Leaderboard
Discussions
Submitted 2 days ago • Score: 10.00
Status: Accepted
Test Case #0
Test Case #1
Test Case #2
Test Case #3
Test Case #4
Test Case #5
Test Case #6
Test Case #7
Submitted Code
Language: Python 3
Open in editor
if __name__=="__main__":
happiness = 0
n,m = map(int, input().strip() .split(' '))
arr = list(map(int, input().strip().split(' ')))
good = set(map(int, input().strip().split(' ')))
bad = set(map(int, input().strip().split(' ')))
if (n>=1 and n<=100000) and (m>=1 and m<=100000):
for i in arr:
if i in good:
happiness += 1
elif i in bad:
happiness -= 1
print(happiness)
else:
print('Wrong input')


HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
sreeanand40
All Contests › AI SIG CONTEST › Triangle Quest 2
Triangle Quest 2
Problem
Submissions
Leaderboard
Discussions
Submitted 2 days ago • Score: 10.00
Status: Accepted
Test Case #0
Test Case #1
Test Case #2
Test Case #3
Test Case #4
Test Case #5
Submitted Code
Language: Python 3
Open in editor
for i in range(1,int(input())+1):
print(((pow(10,i)-1)//9)**2)


HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
sreeanand40
All Contests > AI SIG CONTEST > Validating Credit Card Numbers
Validating Credit Card Numbers
Problem
Submissions
Leaderboard
Discussions
Submitted 2 days ago • Score: 10.00
Status: Accepted
Test Case #0
Test Case #1
Test Case #2
Test Case #3
Test Case #4
Test Case #5
Submitted Code
Language: Python 3
Open in editor
import re
# taking input from user
n = int(input())
for t in range(n):
#taking the credit card number from user
credit = input().strip()
credit_removed_hiphen = credit.replace('-','')
# valid is true in the beggining
valid = True
length_16 = bool(re.match(r'^[4-6]\d{15}$',credit))
length_19 = bool(re.match(r'^[4-6]\d{3}-\d{4}-\d{4}-\d{4}$',credit))
consecutive = bool(re.findall(r'(?=(\d)\1\1)',credit_removed_hiphen))
# checking if the above expressions are true
if length_16 == True or length_19 == True:
if consecutive == True:
valid=False
else:
valid = False
if valid == True:
print('Valid')
else:
print('Invalid')