AI at ACM

AI-ACM SIG Tasks

Name : Swathy Rajesh

Roll no. : AM.EN.U4AIE21063

Branch : S3 CSE-AI (Batch A)

TASK

Category 2

1) **Complete Introduction to Machine Learning course on Kaggle:**
   Time taken: 6 hours max
   Link: https://www.kaggle.com/learn/intro-to-machine-learning
2) **Hackerrank Contest:**
   Time taken: 5 days max
   Link: www.hackerrank.com/ai-sig-contest

HACKERRANK – TASK 2

Problem – 1

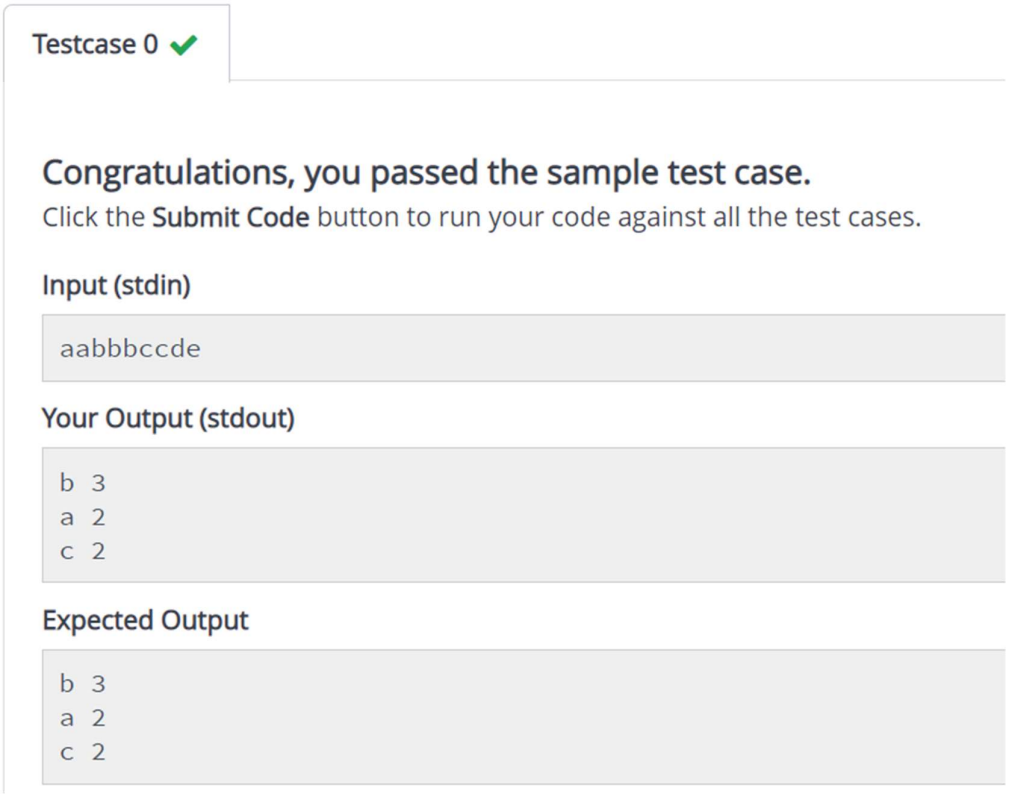
COMPANY LOGO

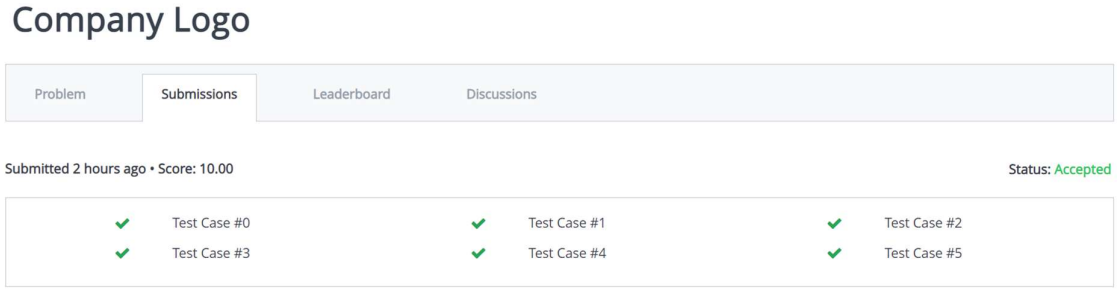
CODE:

```
import math
import os
import random
import re
import sys
from collections import Counter
if __name__ == '__main__':
    s = input()
    z = Counter(s)
    z = Counter(sorted(s)).most_common(3)
```

```
for x in z:
    print(*x)
```

OUTPUT:





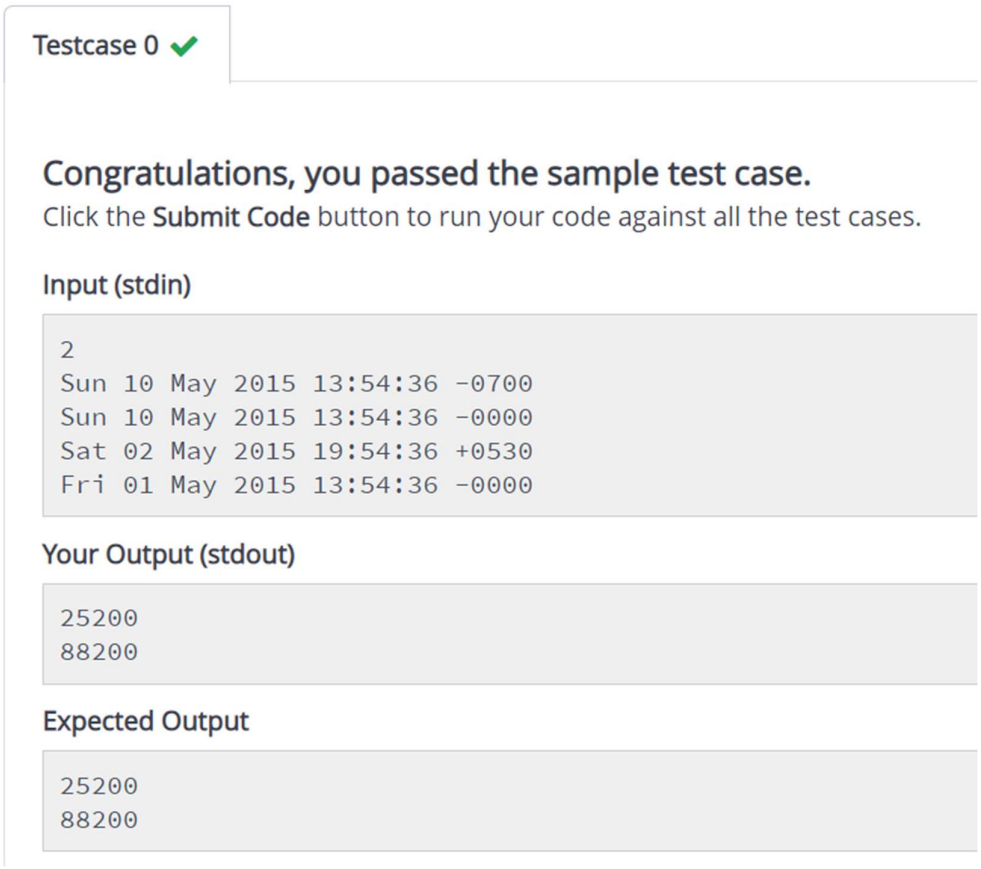
Problem- 2

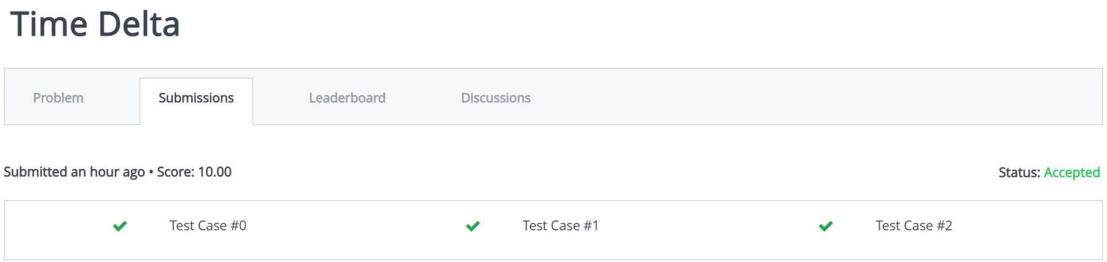
TIME DELTA

CODE:

```
import math
import os
import random
import re
import sys
from datetime import datetime
# Complete the time_delta function below.
def time_delta(t1, t2):
    time_1 = datetime.strptime(t1, '%a %d %b %Y %H:%M:%S %z')
    time_2 = datetime.strptime(t2, '%a %d %b %Y %H:%M:%S %z')
    return str(int(abs(time_1-time_2).total_seconds()))
if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
    t = int(input())
    for t_itr in range(t):
        t1 = input()
        t2 = input()
        delta = time_delta(t1, t2)
        fptr.write(delta + '\n')
    fptr.close()
```

OUTPUT:





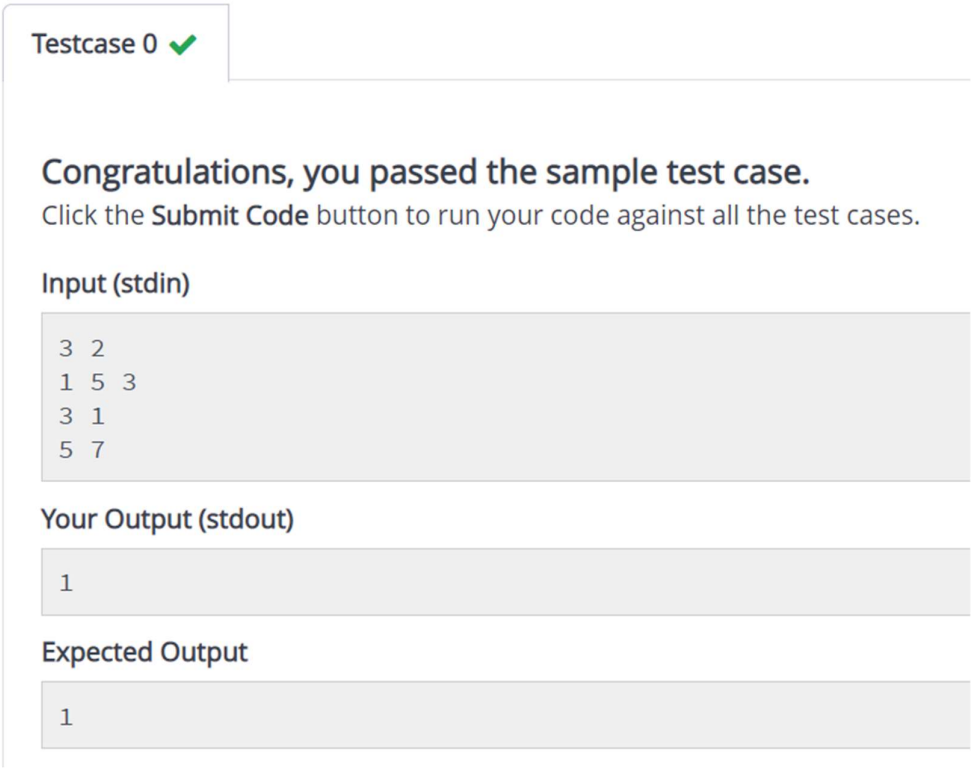
Problem – 3
NO IDEA!

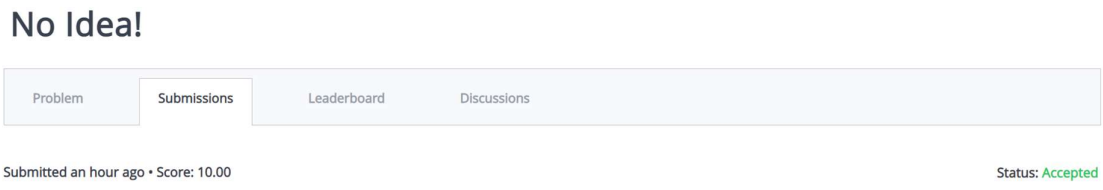
<u>CODE:</u>

```
n,m = list(map(int,input().split()))
elements_array = list(map(int,input().split()))
A = set(map(int,input().split()))
B = set(map(int,input().split()))
happiness = 0
for i in elements_array:
    if i in A:
```

```
        happiness += 1
    if i in B:
        happiness -= 1
print(happiness)
```

OUTPUT:



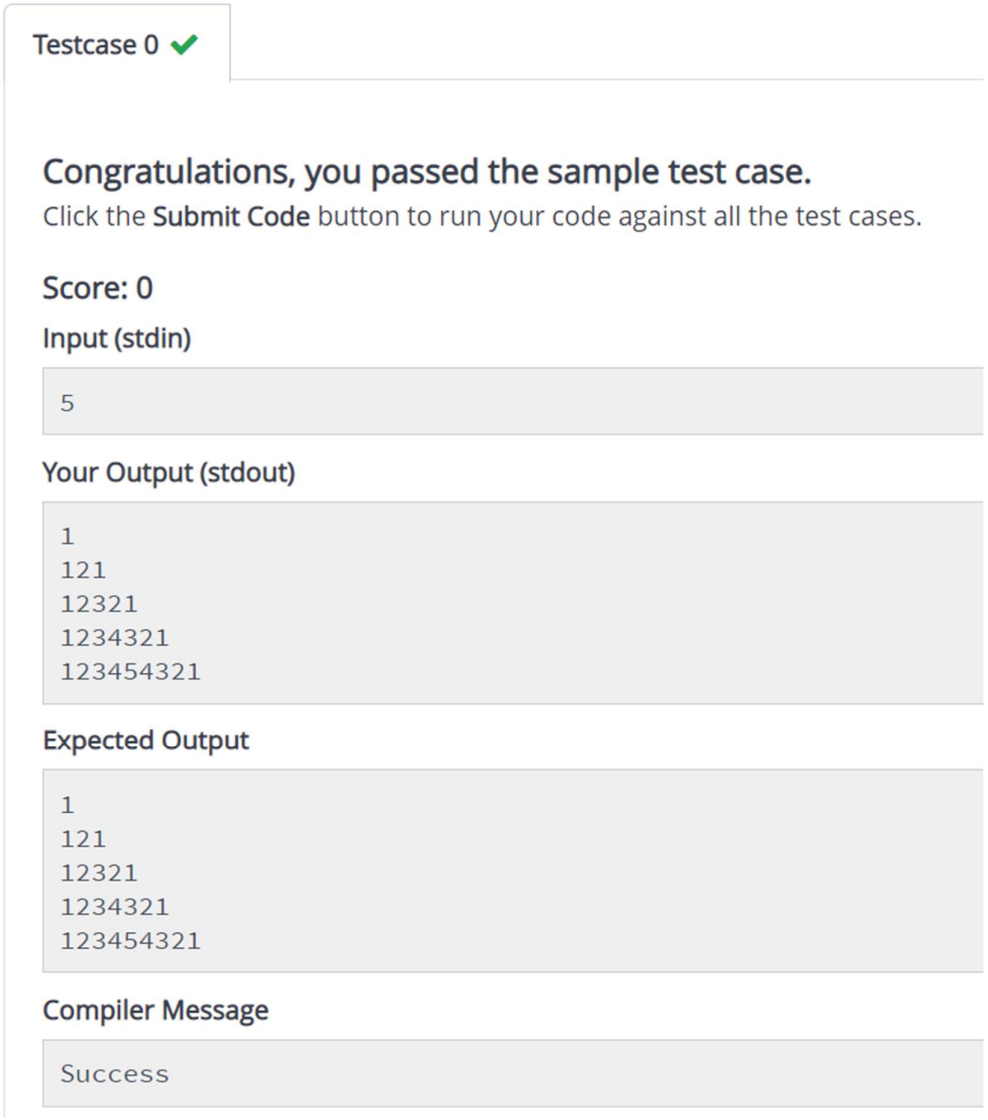


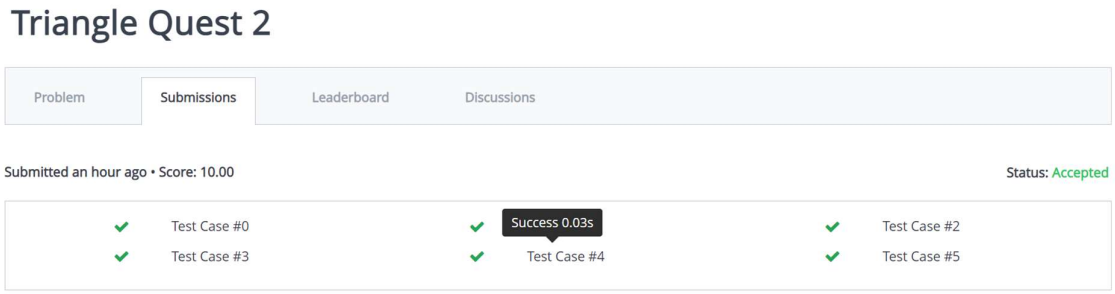
Problem 4:
TRIANGLE QUEST 2

CODE:

```
for i in range(1,int(input())+1):
    print (((10 ** i)//9)**2)
```

OUTPUT:



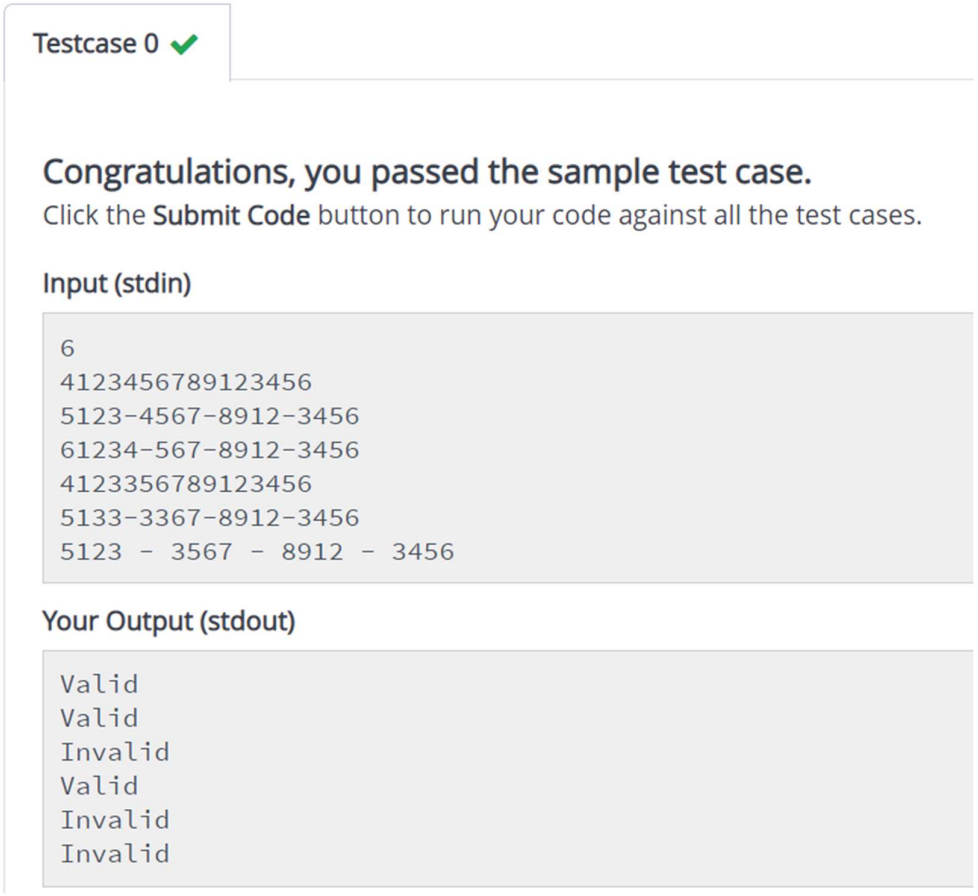


Problem 5:
VALIDATING CREDIT CARD NUMBERS

CODE:
import re

```
for i in range (int(input())):
    N = input()
    t = re.search(r"^[456]\d{15}$|^[456]\d{3}-\d{4}-\d{4}-\d{4}$",N)
    if(t):
        if(re.search(r"(\d)\1{3,}|(\d)\2{1}-(\d)\2{1}|-(\d)\4{3}-",N)):
            print("Invalid")
        else:
            print("Valid")
    else:
        print("Invalid")
```

OUTPUT:



Expected Output

```
Valid
Valid
Invalid
Valid
Invalid
Invalid
```

Validating Credit Card Numbers

Problem | Submissions | Leaderboard | Discussions

Submitted 34 minutes ago • Score: 10.00 Status: Accepted

✔ Test Case #0	✔ Test Case #1	✔ Test Case #2
✔ Test Case #3	✔ Test Case #4	✔ Test Case #5

AI SIG CONTEST Details ▶

Challenges

Company Logo
Success Rate: 100.00% Max Score: 10 Difficulty: Medium — Try Again

Time Delta
Success Rate: 100.00% Max Score: 10 Difficulty: Medium — Try Again

No Idea!
Success Rate: 100.00% Max Score: 10 Difficulty: Medium — Try Again

Triangle Quest 2
Success Rate: 100.00% Max Score: 10 Difficulty: Medium — Try Again

Validating Credit Card Numbers
Success Rate: 75.00% Max Score: 10 Difficulty: Medium — Try Again