CE-860 AI Assignment

HackerRank Tasks

Muhammad Usman Abbas

450498

RIME23

Aim:

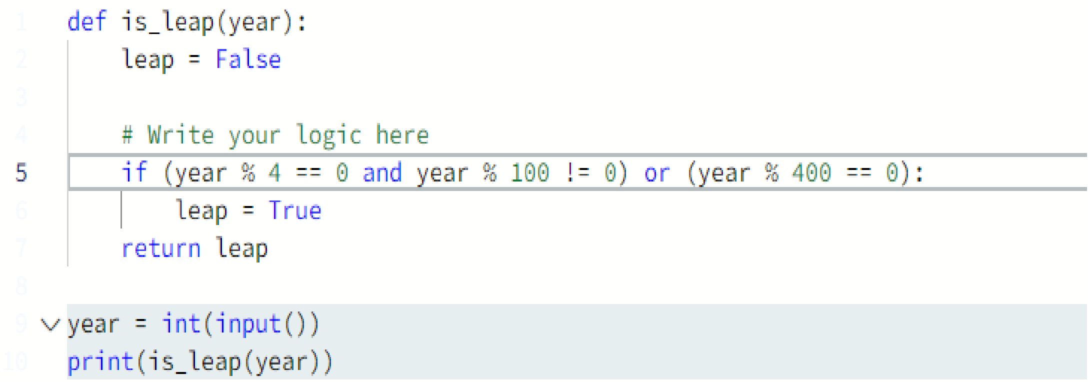
Complete Medium and Hard tasks in HackerRank website.

Synopsis:

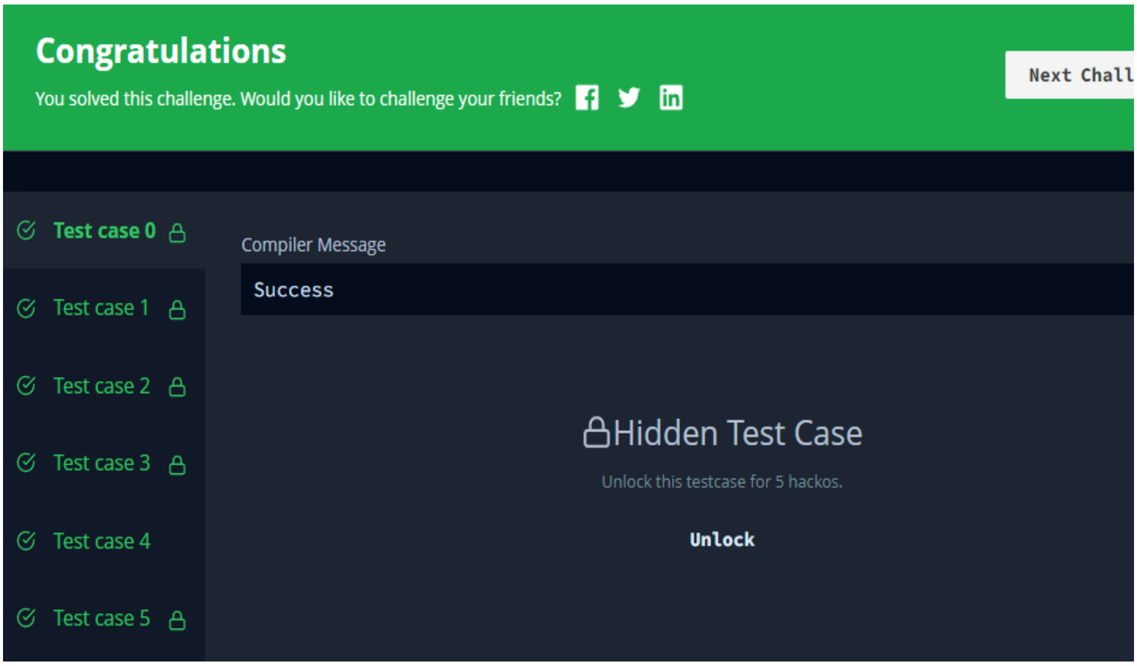
I first completed the Easy tasks and built on the already knowledge I had of python. Refreshed some earlier concepts and focused new logic building while expanding my understanding of functions. I faced difficulty in completing even the medium task and took help form W3Schools, GeeksforGeeks, PythonDocs and OpenAI. Hard tasks took a lot of time and in the end, I had to resort to using OpenAI, although I did make an effort to understand how it solved the problem. Following are the solved code and respective outputs:

Medium Tasks

1. **Write a Function**



```
def is_leap(year):
    leap = False

    # Write your logic here
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        leap = True
    return leap

year = int(input())
print(is_leap(year))
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

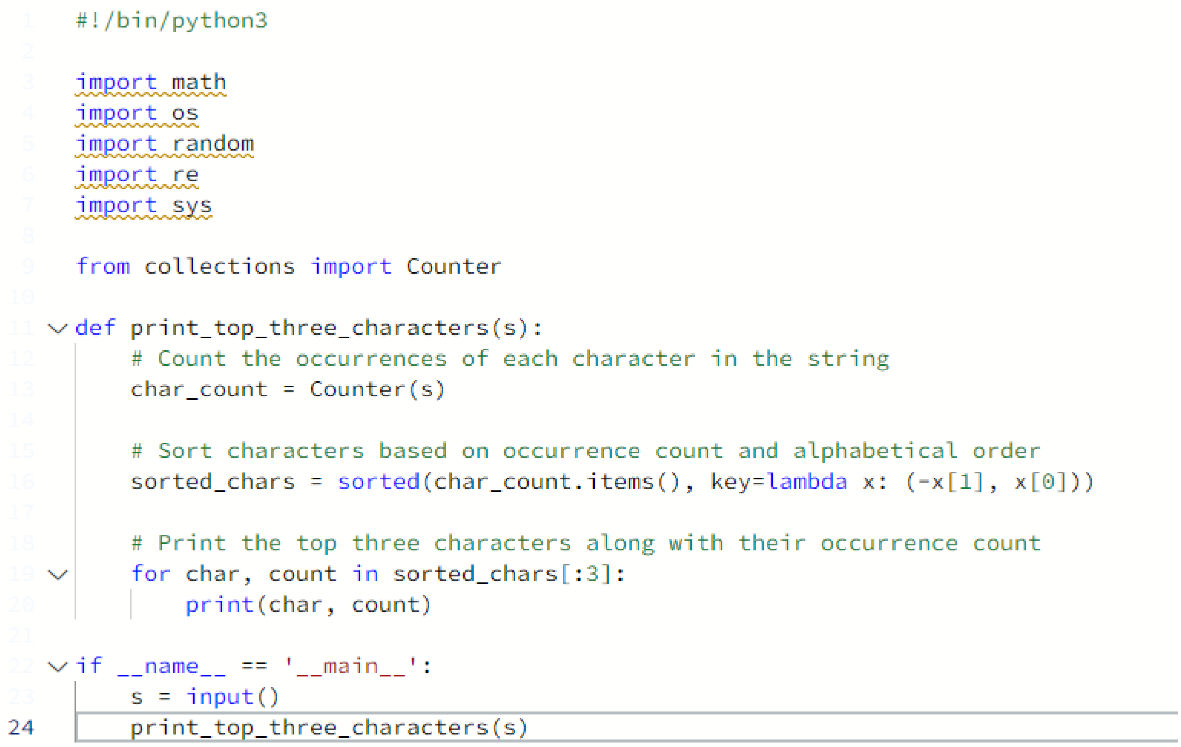
Compiler Message

```
Success
```

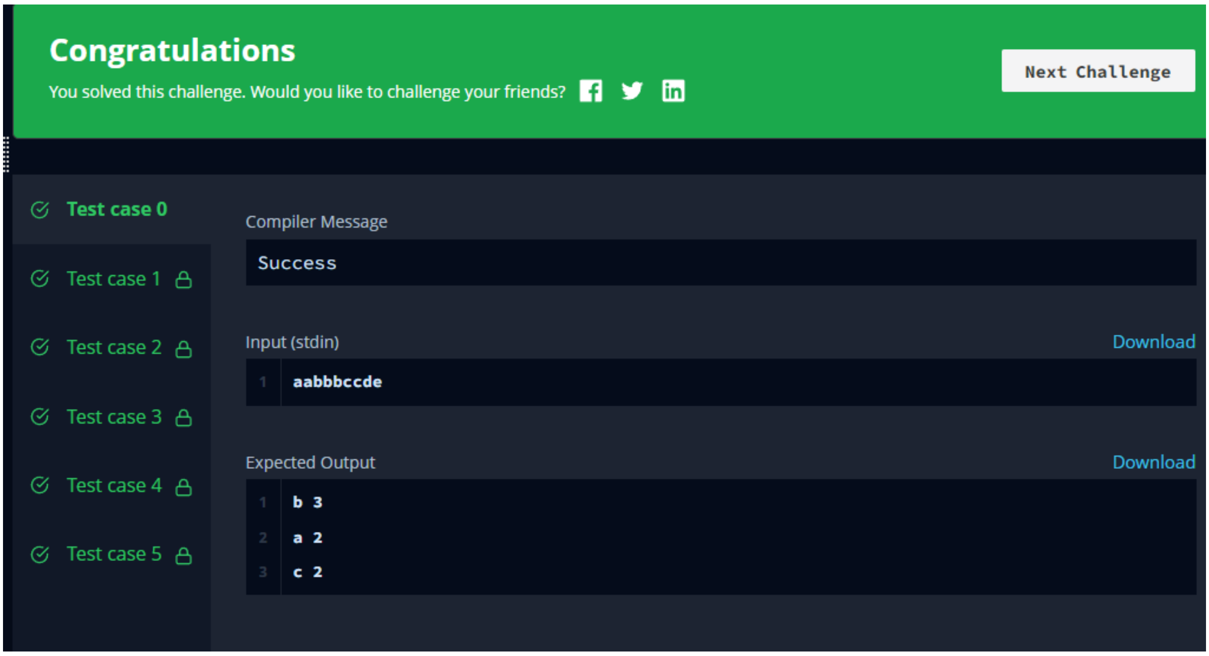
Hidden Test Case

Unlock this testcase for 5 hackos.

2. Company Logo



```python
#!/bin/python3

import math
import os
import random
import re
import sys

from collections import Counter

def print_top_three_characters(s):
    # Count the occurrences of each character in the string
    char_count = Counter(s)

    # Sort characters based on occurrence count and alphabetical order
    sorted_chars = sorted(char_count.items(), key=lambda x: (-x[1], x[0]))

    # Print the top three characters along with their occurrence count
    for char, count in sorted_chars[:3]:
        print(char, count)

if __name__ == '__main__':
    s = input()
    print_top_three_characters(s)
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5

Compiler Message

```
Success
```

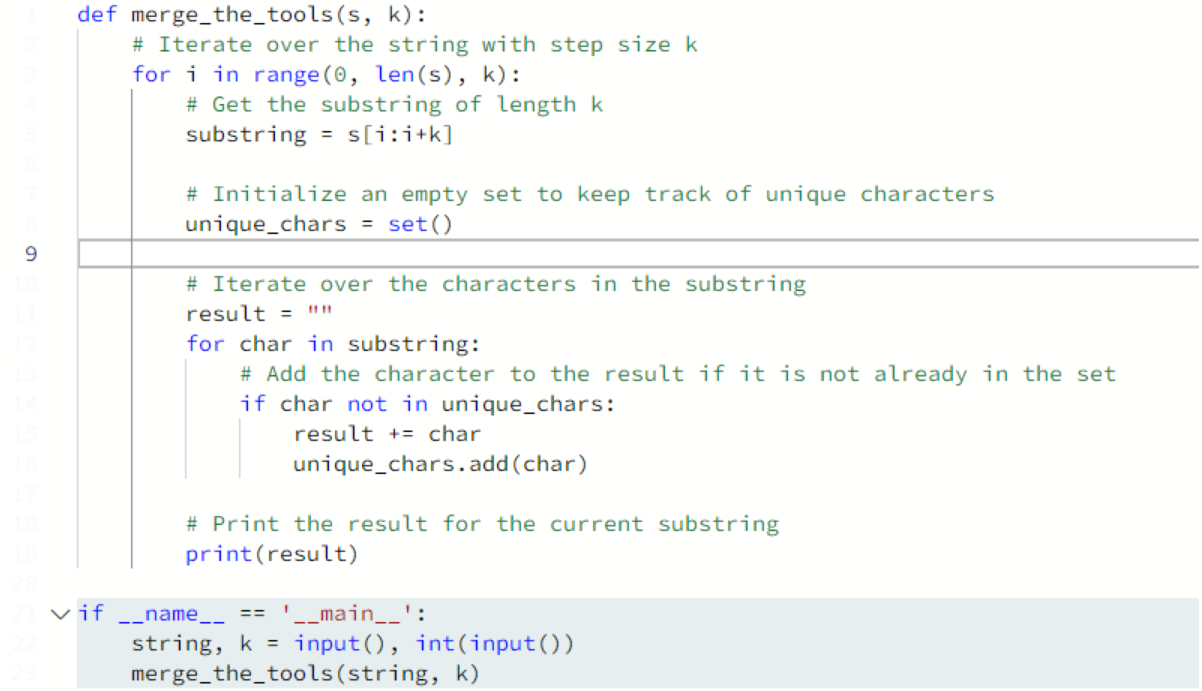
Input (stdin)

```
aabbbccde
```

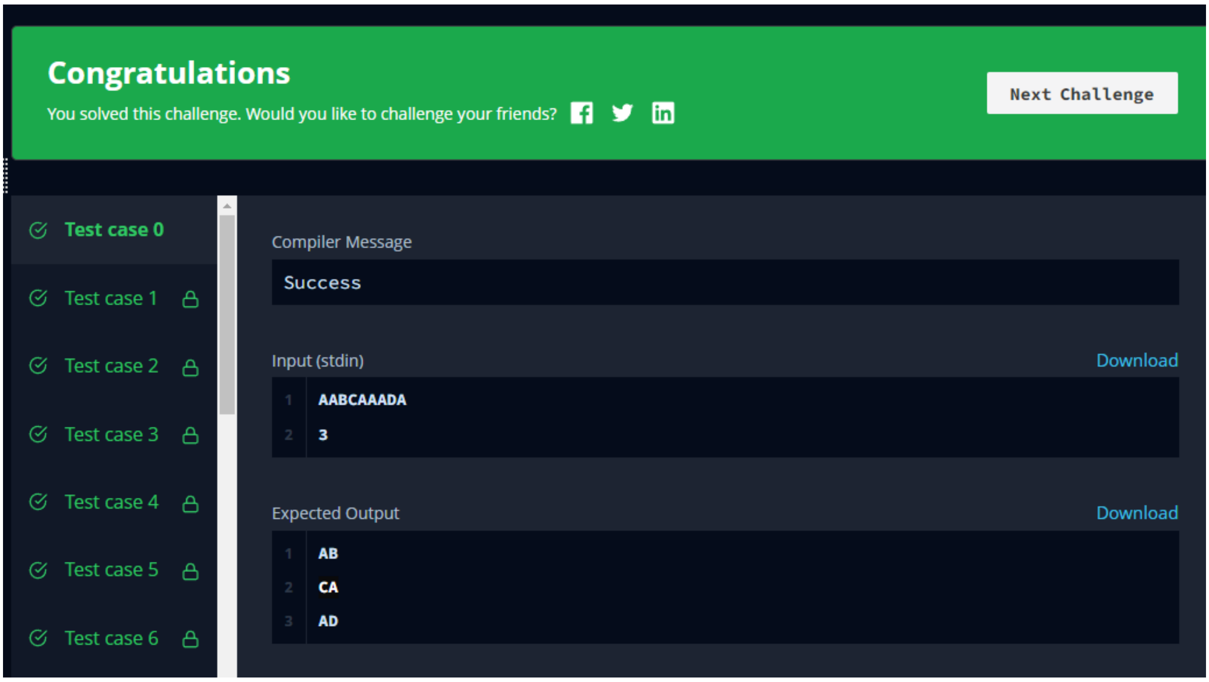
Expected Output

```
b 3
a 2
c 2
```

3. Merge the Tools



```python
def merge_the_tools(s, k):
    # Iterate over the string with step size k
    for i in range(0, len(s), k):
        # Get the substring of length k
        substring = s[i:i+k]
        
        # Initialize an empty set to keep track of unique characters
        unique_chars = set()
        
        # Iterate over the characters in the substring
        result = ""
        for char in substring:
            # Add the character to the result if it is not already in the set
            if char not in unique_chars:
                result += char
                unique_chars.add(char)
        
        # Print the result for the current substring
        print(result)

if __name__ == '__main__':
    string, k = input(), int(input())
    merge_the_tools(string, k)
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

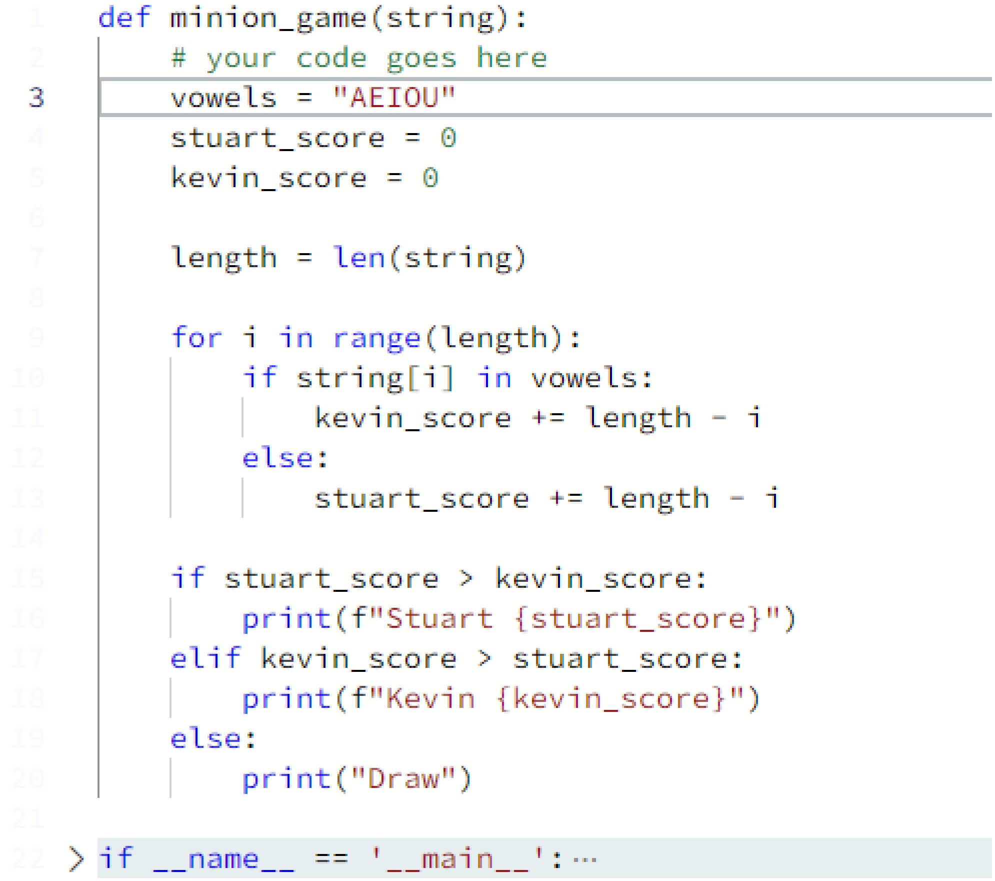
Input (stdin)

```
AABCAAADA
3
```

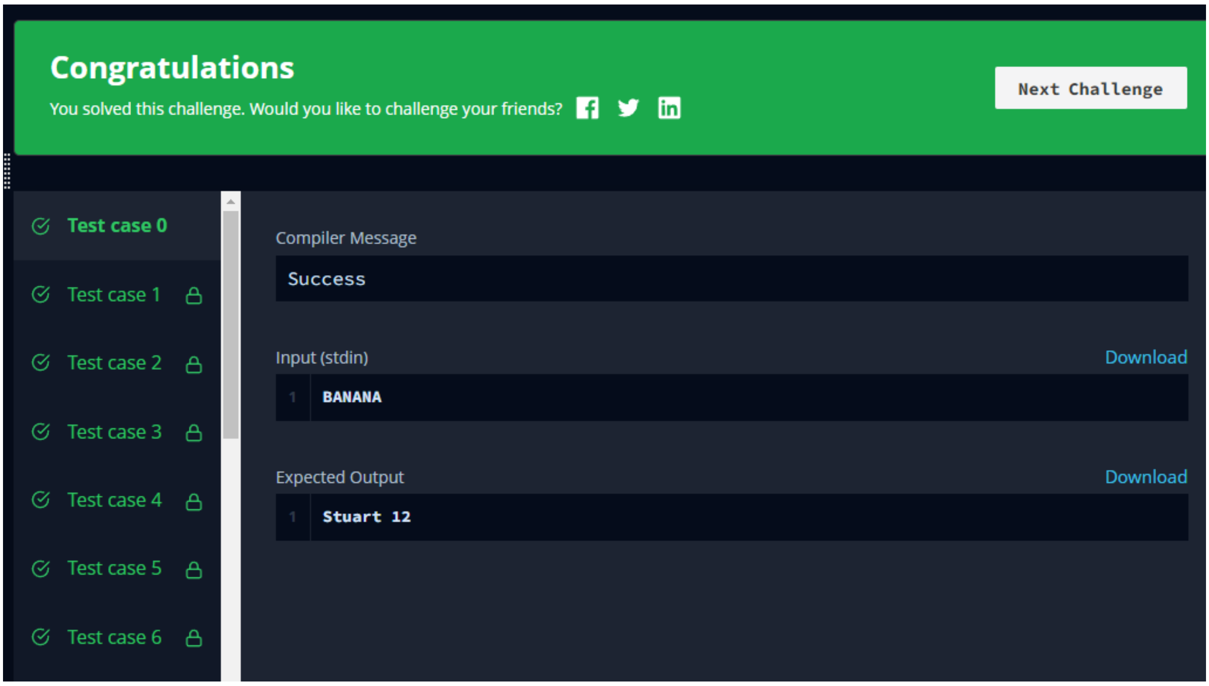
Expected Output

```
AB
CA
AD
```

4. Minion Game



```
def minion_game(string):
    # your code goes here
    vowels = "AEIOU"
    stuart_score = 0
    kevin_score = 0

    length = len(string)

    for i in range(length):
        if string[i] in vowels:
            kevin_score += length - i
        else:
            stuart_score += length - i

    if stuart_score > kevin_score:
        print(f"Stuart {stuart_score}")
    elif kevin_score > stuart_score:
        print(f"Kevin {kevin_score}")
    else:
        print("Draw")

if __name__ == '__main__':…
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

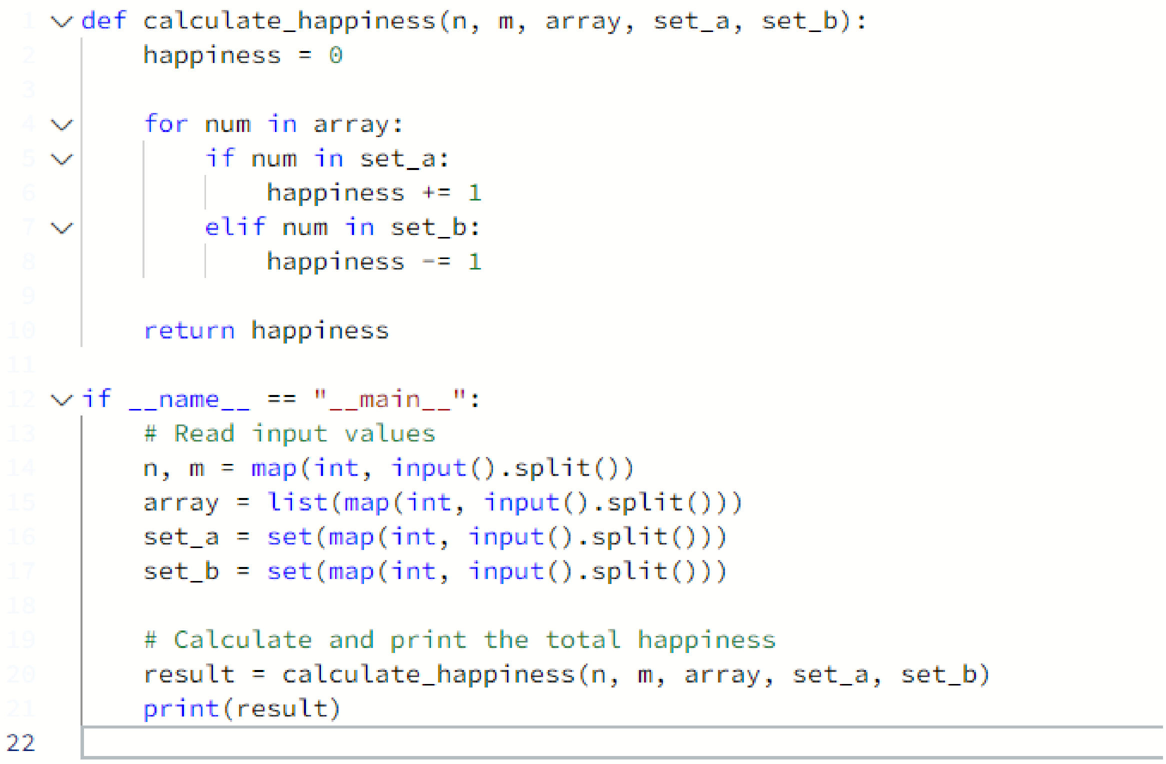
Input (stdin)

```
BANANA
```

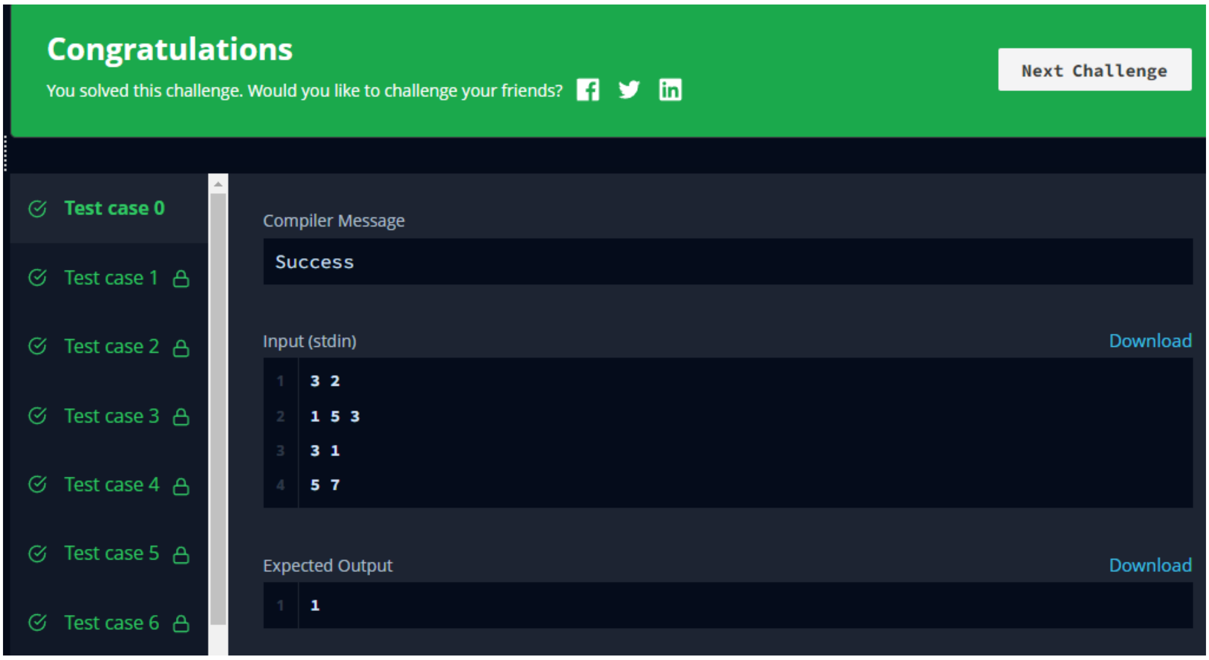
Expected Output

```
Stuart 12
```

5. No Idea



```python
def calculate_happiness(n, m, array, set_a, set_b):
    happiness = 0

    for num in array:
        if num in set_a:
            happiness += 1
        elif num in set_b:
            happiness -= 1

    return happiness

if __name__ == "__main__":
    # Read input values
    n, m = map(int, input().split())
    array = list(map(int, input().split()))
    set_a = set(map(int, input().split()))
    set_b = set(map(int, input().split()))

    # Calculate and print the total happiness
    result = calculate_happiness(n, m, array, set_a, set_b)
    print(result)
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

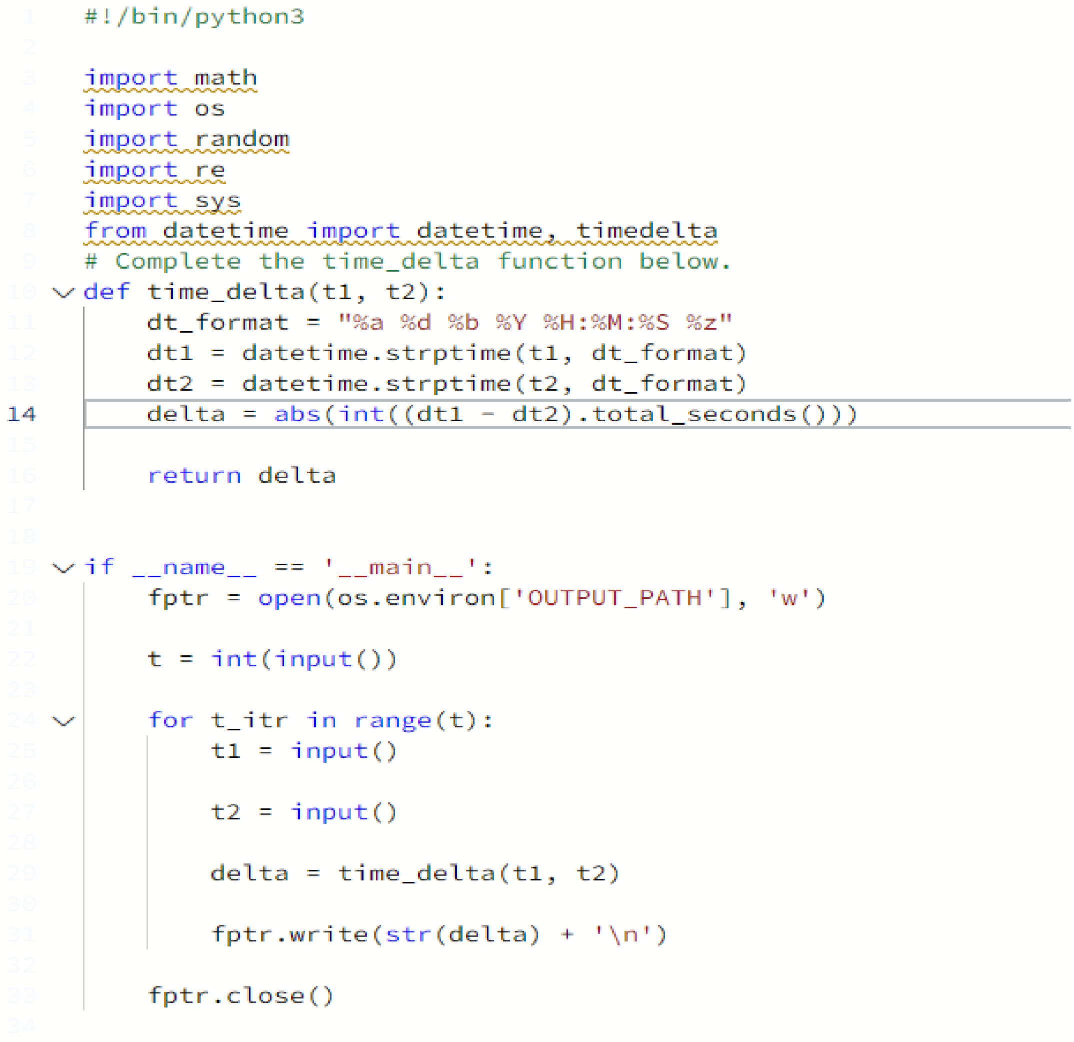
Input (stdin)

```
3 2
1 5 3
3 1
5 7
```

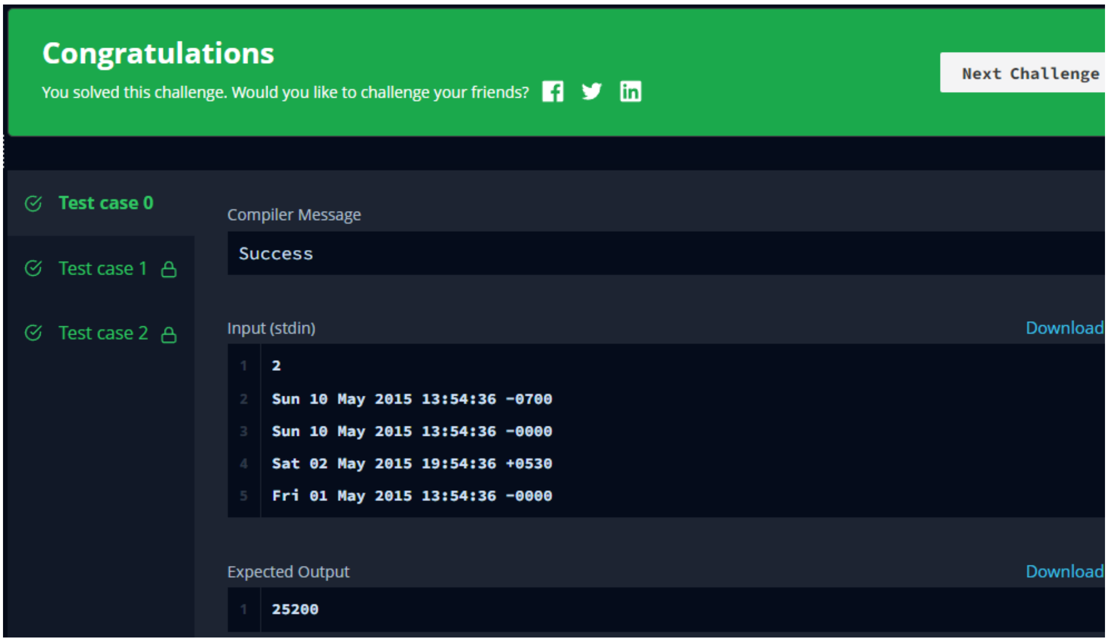
Expected Output

```
1
```

6. Time Delta



```
#!/bin/python3

import math
import os
import random
import re
import sys
from datetime import datetime, timedelta
# Complete the time_delta function below.
def time_delta(t1, t2):
    dt_format = "%a %d %b %Y %H:%M:%S %z"
    dt1 = datetime.strptime(t1, dt_format)
    dt2 = datetime.strptime(t2, dt_format)
    delta = abs(int((dt1 - dt2).total_seconds()))

    return delta


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(str(delta) + '\n')

    fptr.close()
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

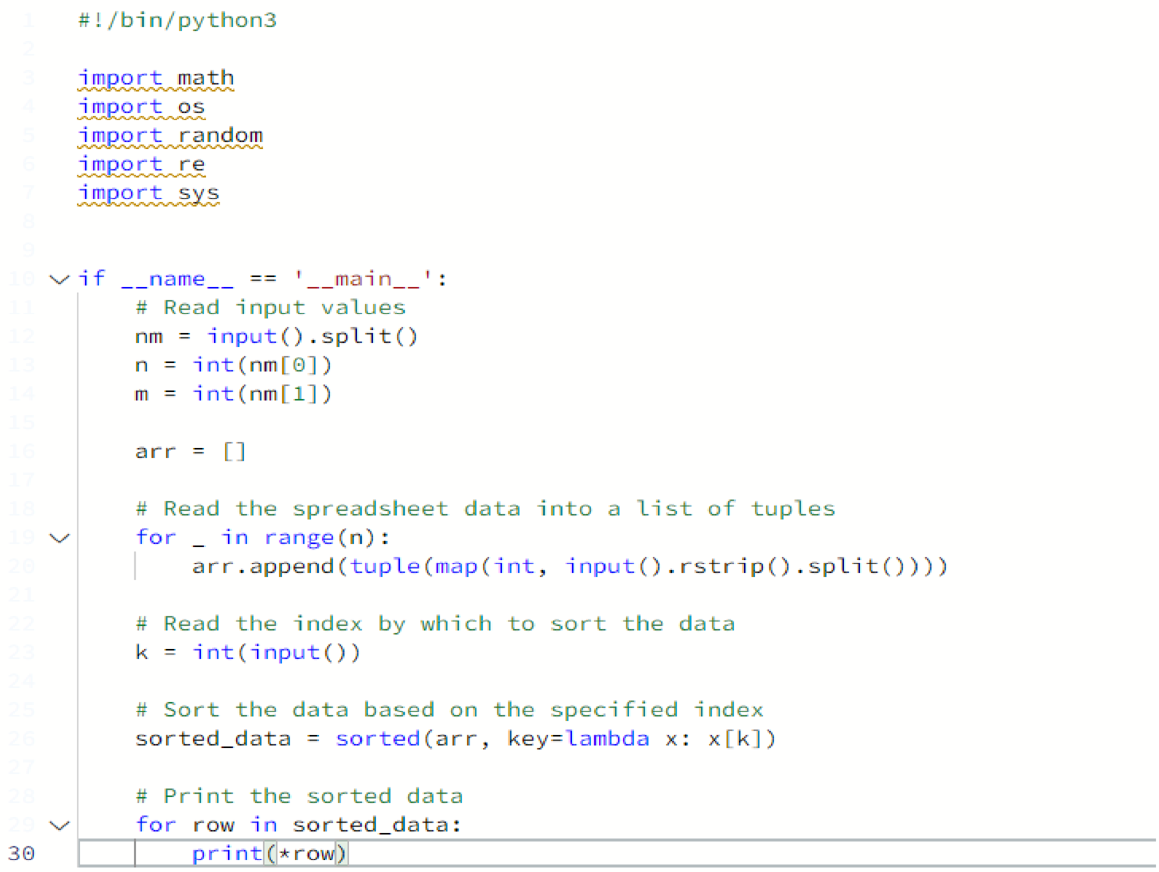
Input (stdin)

```
2
Sun 10 May 2015 13:54:36 -0700
Sun 10 May 2015 13:54:36 -0000
Sat 02 May 2015 19:54:36 +0530
Fri 01 May 2015 13:54:36 -0000
```

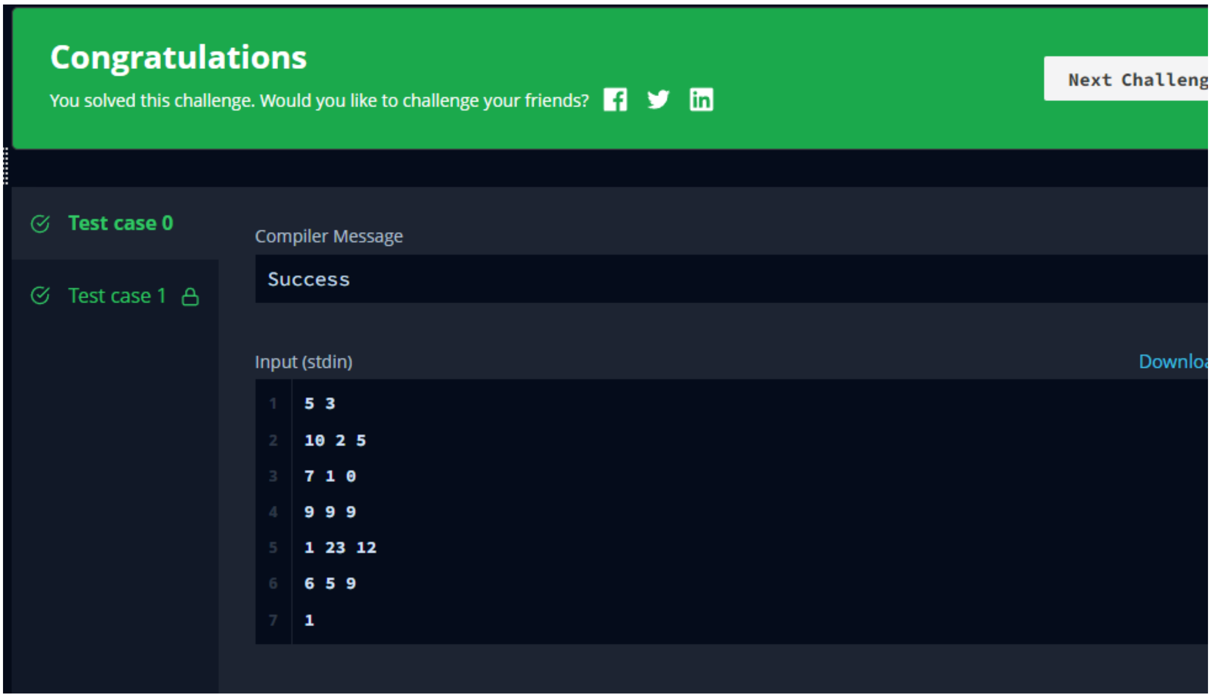
Expected Output

```
25200
```

7. Athlete



```
#!/bin/python3

import math
import os
import random
import re
import sys


if __name__ == '__main__':
    # Read input values
    nm = input().split()
    n = int(nm[0])
    m = int(nm[1])

    arr = []

    # Read the spreadsheet data into a list of tuples
    for _ in range(n):
        arr.append(tuple(map(int, input().rstrip().split())))

    # Read the index by which to sort the data
    k = int(input())

    # Sort the data based on the specified index
    sorted_data = sorted(arr, key=lambda x: x[k])

    # Print the sorted data
    for row in sorted_data:
        print(*row)
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

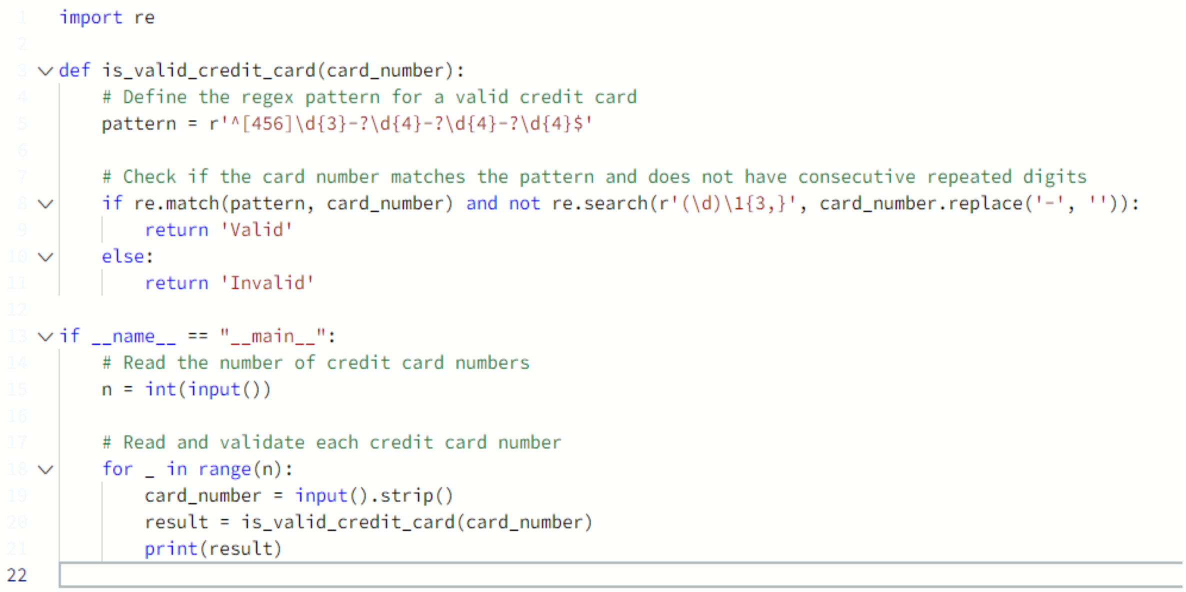
Compiler Message

```
Success
```

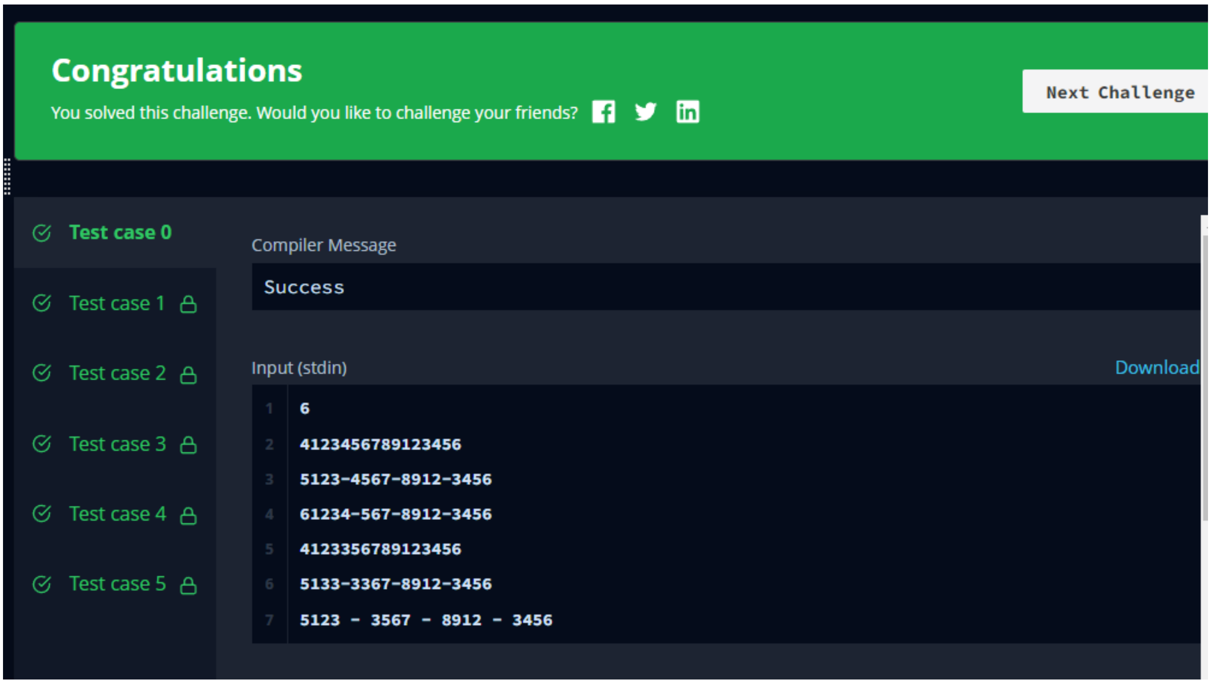
Input (stdin)

```
5 3
10 2 5
7 1 0
9 9 9
1 23 12
6 5 9
1
```

8. Validating Credit Card Numbers



```python
import re

def is_valid_credit_card(card_number):
    # Define the regex pattern for a valid credit card
    pattern = r'^[456]\d{3}-?\d{4}-?\d{4}-?\d{4}$'

    # Check if the card number matches the pattern and does not have consecutive repeated digits
    if re.match(pattern, card_number) and not re.search(r'(\d)\1{3,}', card_number.replace('-', '')):
        return 'Valid'
    else:
        return 'Invalid'

if __name__ == "__main__":
    # Read the number of credit card numbers
    n = int(input())

    # Read and validate each credit card number
    for _ in range(n):
        card_number = input().strip()
        result = is_valid_credit_card(card_number)
        print(result)

```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

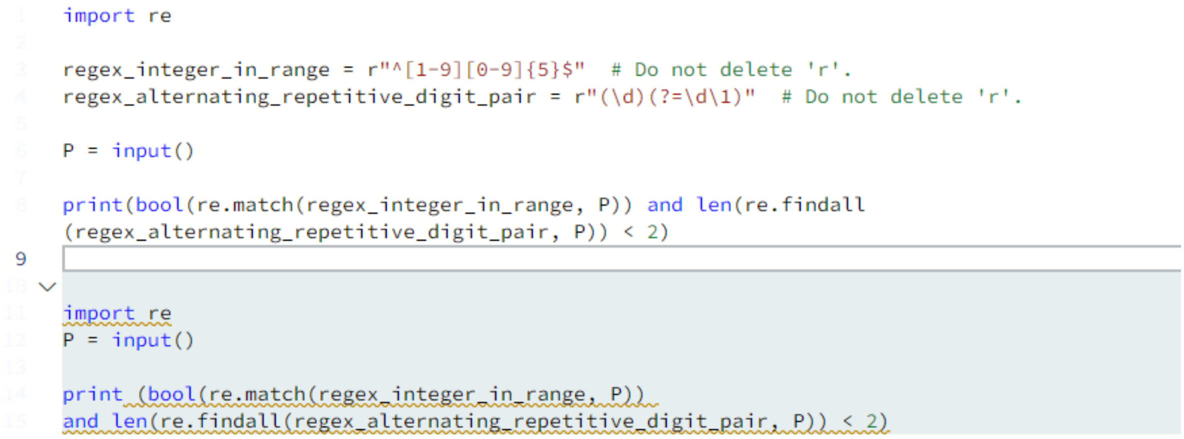
```
Success
```

Input (stdin)

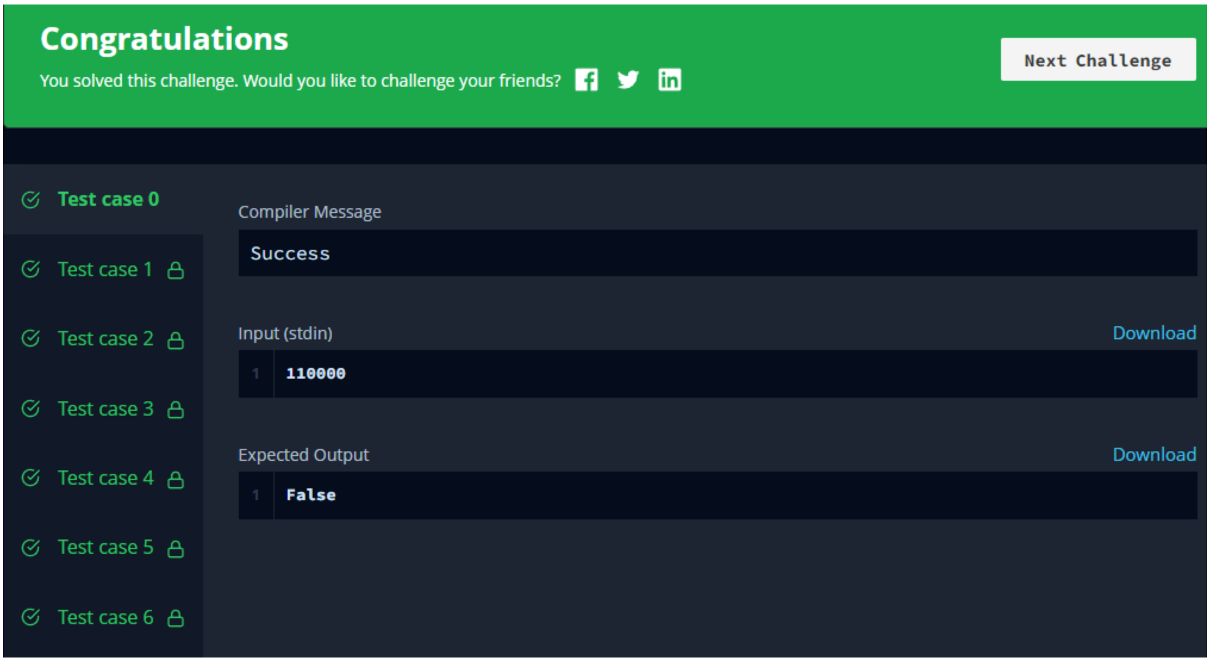
```
6
4123456789123456
5123-4567-8912-3456
61234-567-8912-3456
4123356789123456
5133-3367-8912-3456
5123 - 3567 - 8912 - 3456
```

Hard Tasks

1. Postal Code



```
import re

regex_integer_in_range = r"^[1-9][0-9]{5}$"  # Do not delete 'r'.
regex_alternating_repetitive_digit_pair = r"(\d)(?=\d\1)"  # Do not delete 'r'.

P = input()

print(bool(re.match(regex_integer_in_range, P)) and len(re.findall
(regex_alternating_repetitive_digit_pair, P)) < 2)


import re
P = input()

print (bool(re.match(regex_integer_in_range, P))
and len(re.findall(regex_alternating_repetitive_digit_pair, P)) < 2)
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

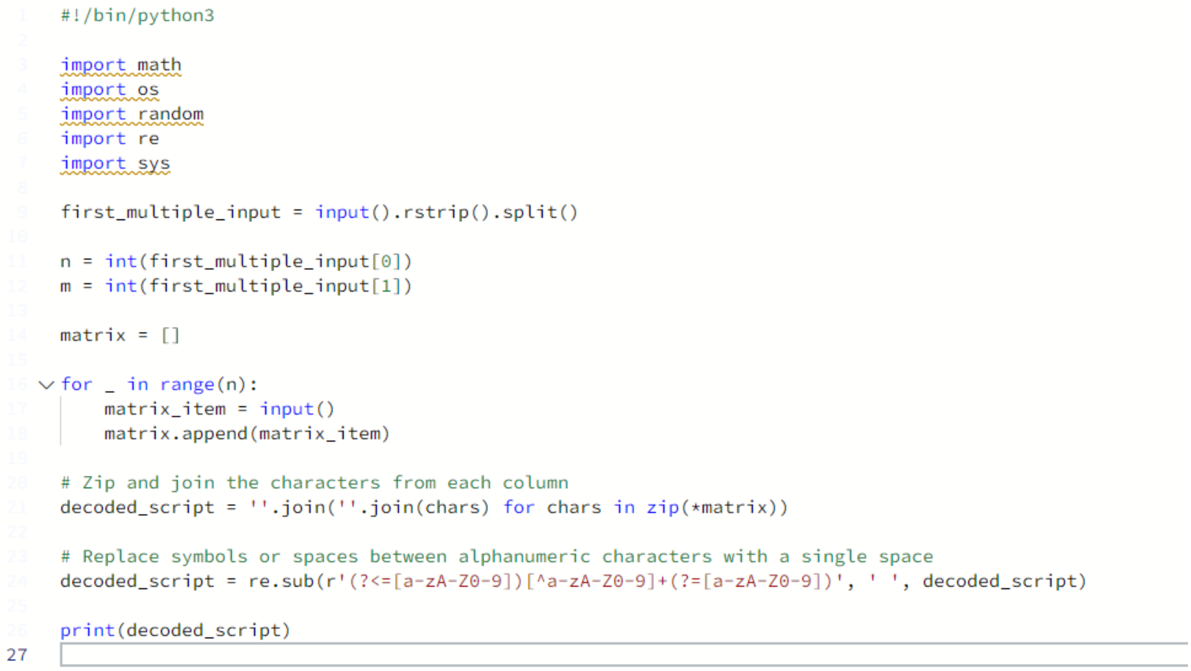
Input (stdin)

```
110000
```

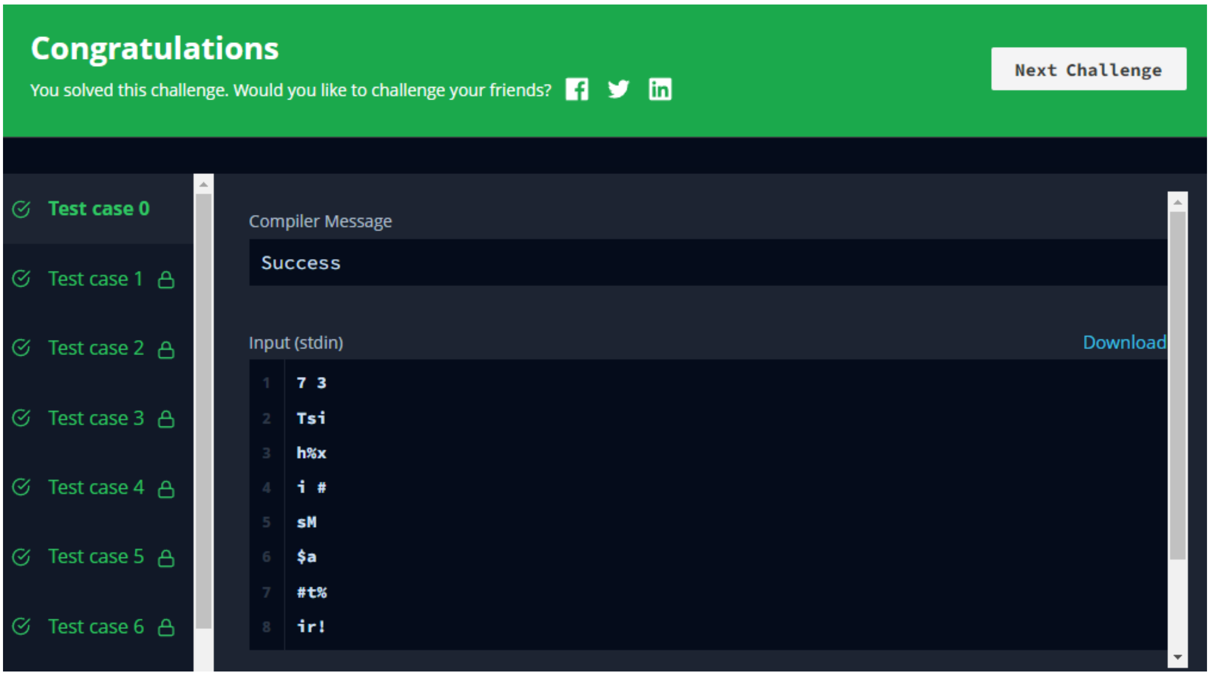
Expected Output

```
False
```

2. Matrix Script



```python
#!/bin/python3

import math
import os
import random
import re
import sys

first_multiple_input = input().rstrip().split()

n = int(first_multiple_input[0])
m = int(first_multiple_input[1])

matrix = []

for _ in range(n):
    matrix_item = input()
    matrix.append(matrix_item)

# Zip and join the characters from each column
decoded_script = ''.join(''.join(chars) for chars in zip(*matrix))

# Replace symbols or spaces between alphanumeric characters with a single space
decoded_script = re.sub(r'(?<=[a-zA-Z0-9])[^a-zA-Z0-9]+(?=[a-zA-Z0-9])', ' ', decoded_script)

print(decoded_script)

```



Compiler Message

```
Success
```

Input (stdin)

```
7 3
Tsi
h%x
i #
sM 
$a 
#t%
ir!
```