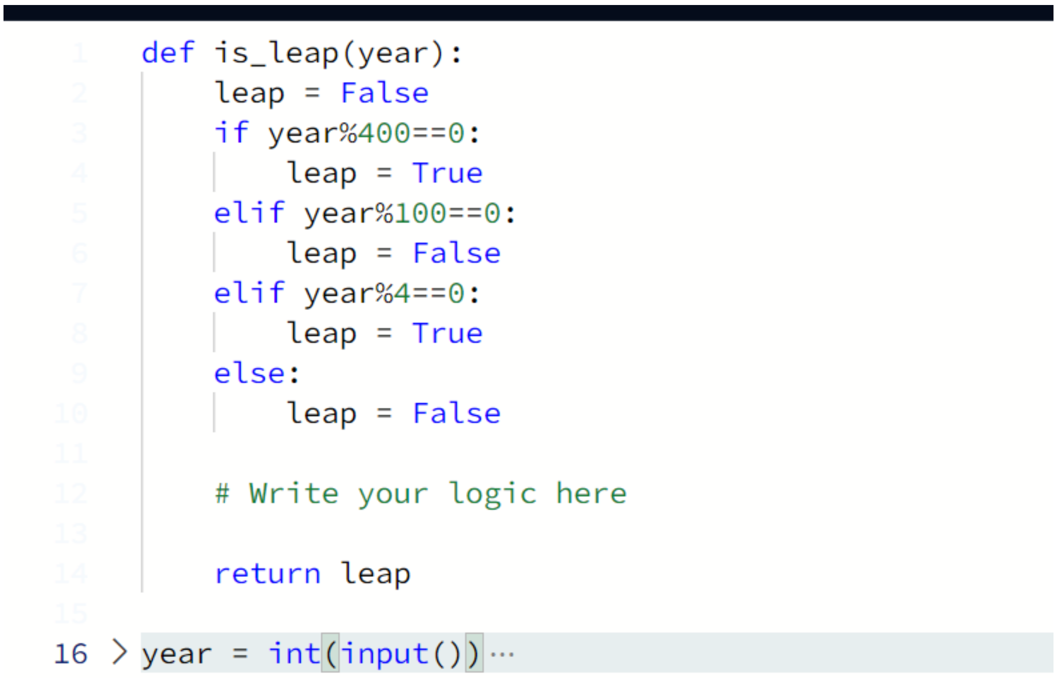
ASSIGNMENT # 3

NAME	:	M. HAMZA ZAMAN
REG NO	:	402581
DEGREE	:	MS (MECHANICAL ENGINEERING)
SUBMITTED TO	:	DR YASAR AYAZ
DATED	:	05 JANUARY 2024

MEDIUM DIFFICULTY CHALLENGES

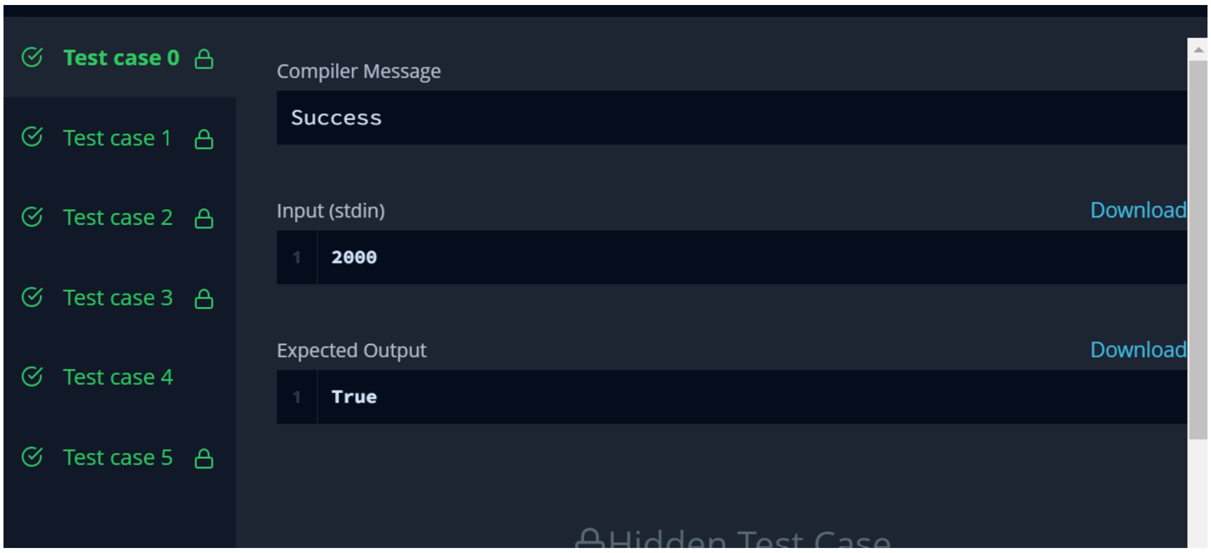
1. **Write a Function**

   Code:



```python
def is_leap(year):
    leap = False
    if year%400==0:
        leap = True
    elif year%100==0:
        leap = False
    elif year%4==0:
        leap = True
    else:
        leap = False

    # Write your logic here

    return leap

year = int(input())···
```

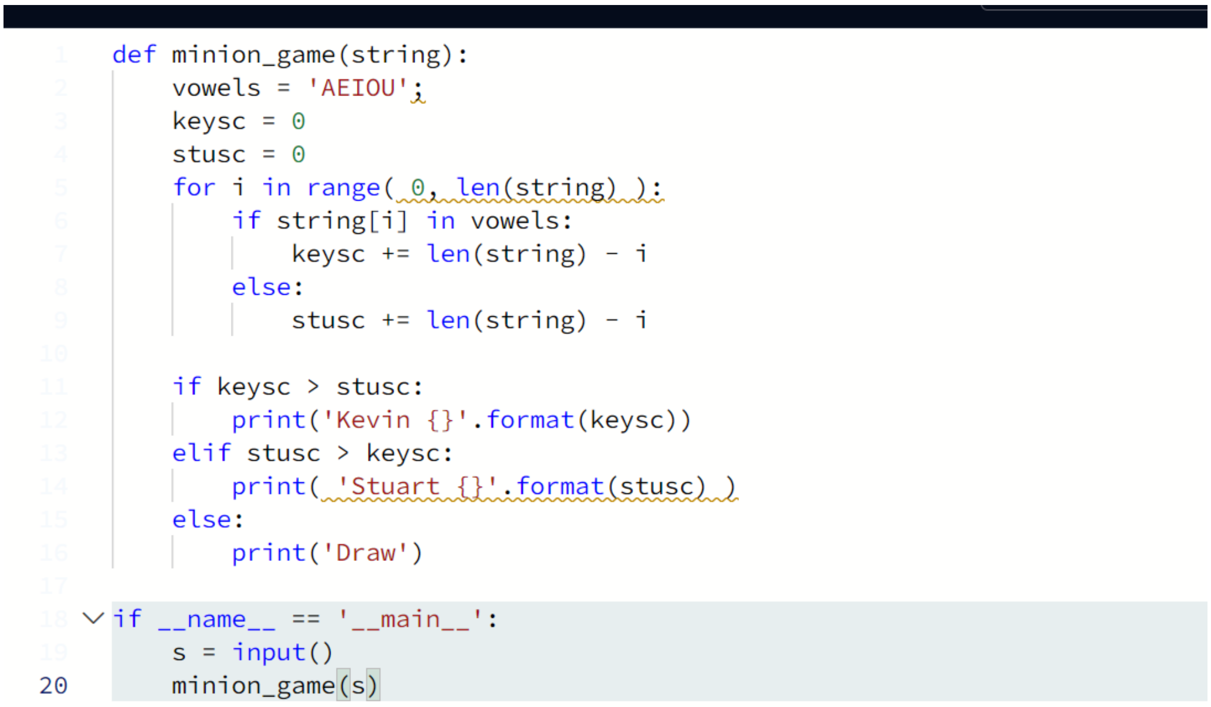
   Output:



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

Compiler Message
Success

Input (stdin)
2000

Expected Output
True

Hidden Test Case
```

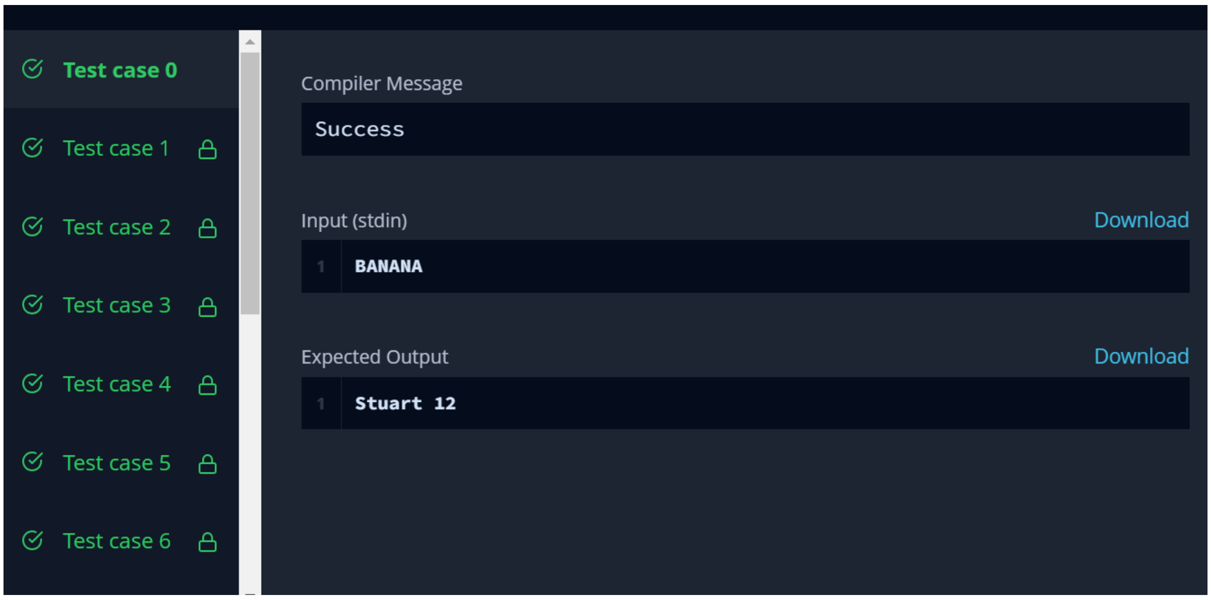
2. The Minion Game

Code:



```python
def minion_game(string):
    vowels = 'AEIOU';
    keysc = 0
    stusc = 0
    for i in range( 0, len(string) ):
        if string[i] in vowels:
            keysc += len(string) - i
        else:
            stusc += len(string) - i

    if keysc > stusc:
        print('Kevin {}'.format(keysc))
    elif stusc > keysc:
        print( 'Stuart {}'.format(stusc) )
    else:
        print('Draw')

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

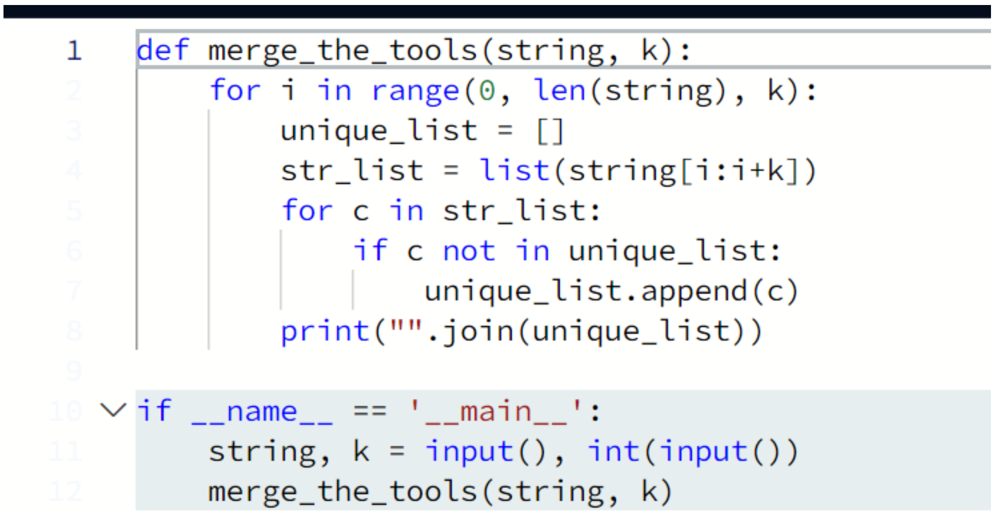
Output:



```
Compiler Message
Success

Input (stdin)
BANANA

Expected Output
Stuart 12
```

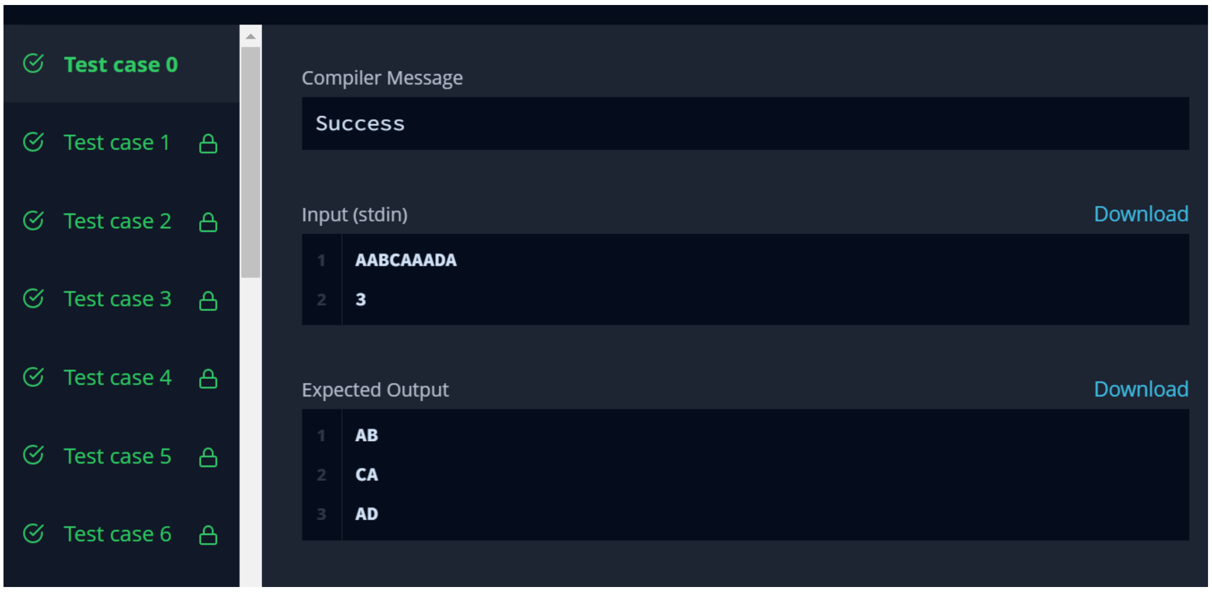
3. **Merge the Tools!**

Code:



```python
def merge_the_tools(string, k):
    for i in range(0, len(string), k):
        unique_list = []
        str_list = list(string[i:i+k])
        for c in str_list:
            if c not in unique_list:
                unique_list.append(c)
        print("".join(unique_list))

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

Output:



Compiler Message

```
Success
```

Input (stdin)

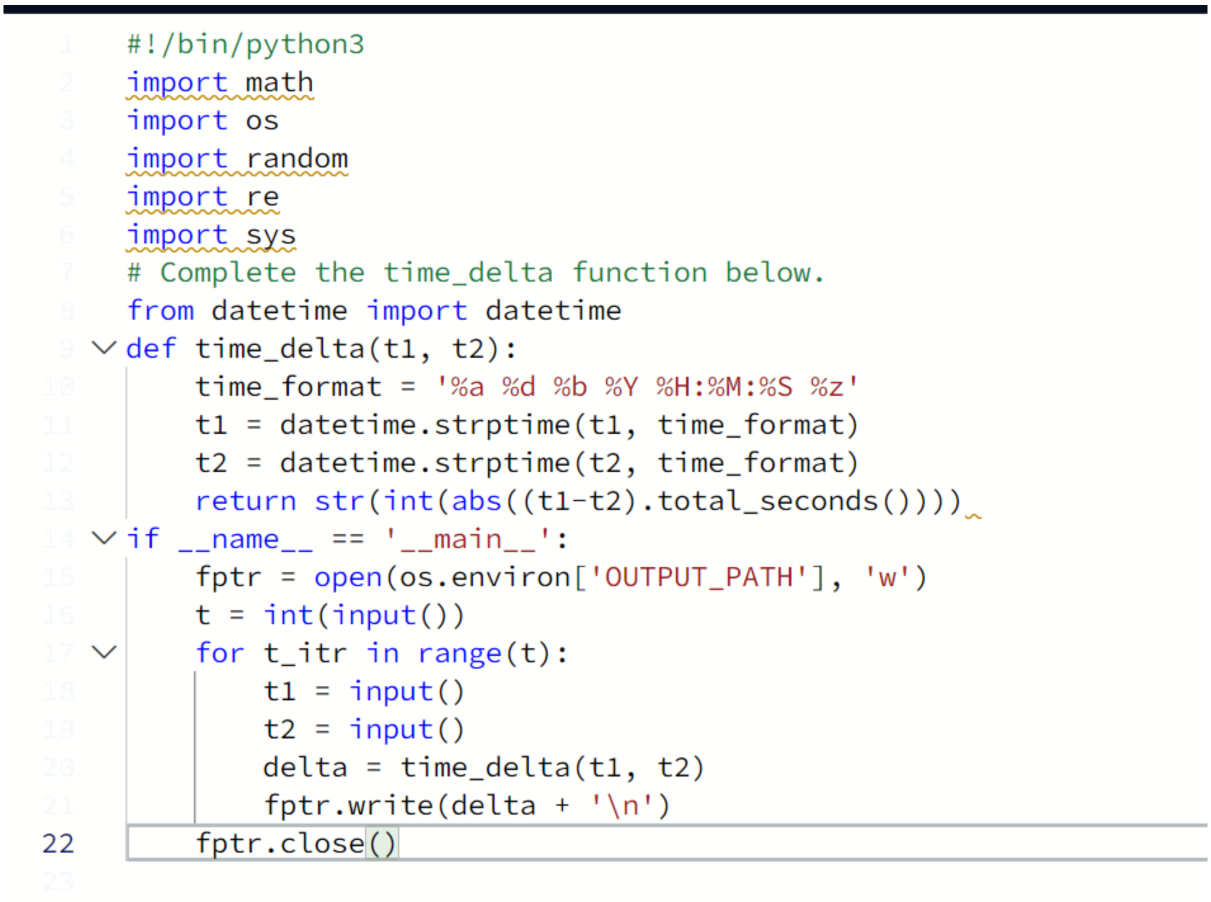
```
AABCAAADA
3
```

Expected Output

```
AB
CA
AD
```

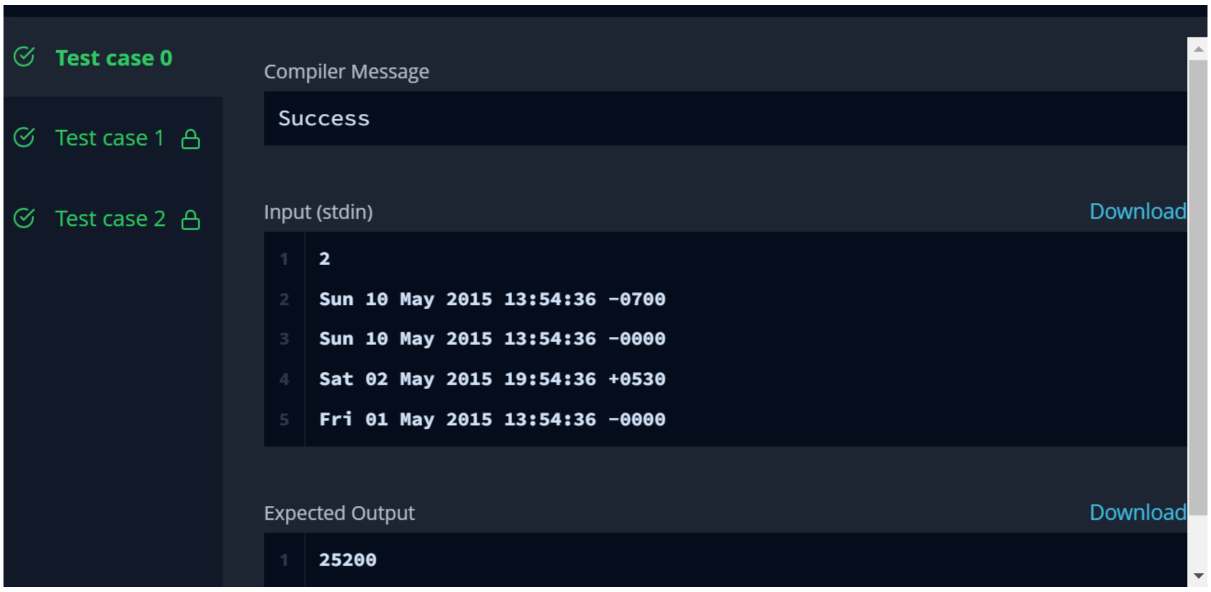
4. Time Delta

Code:



```python
#!/bin/python3
import math
import os
import random
import re
import sys
# Complete the time_delta function below.
from datetime import datetime
def time_delta(t1, t2):
    time_format = '%a %d %b %Y %H:%M:%S %z'
    t1 = datetime.strptime(t1, time_format)
    t2 = datetime.strptime(t2, time_format)
    return str(int(abs((t1-t2).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()
```

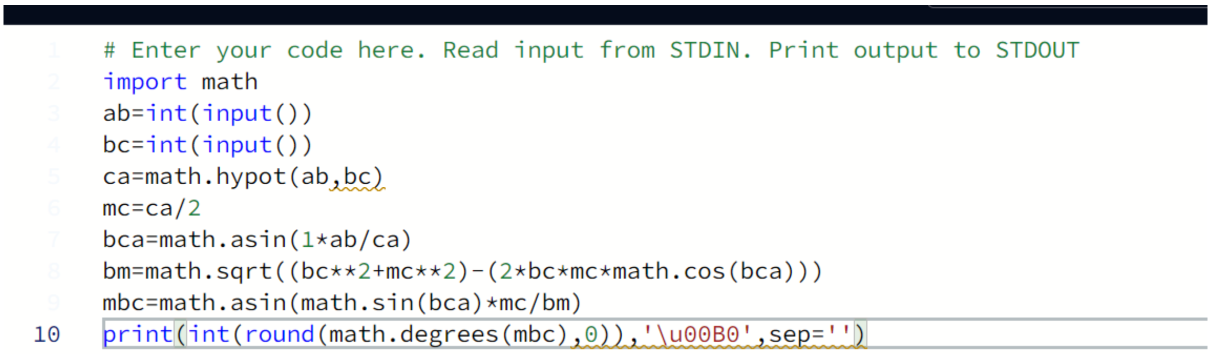
Output:



```
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
```

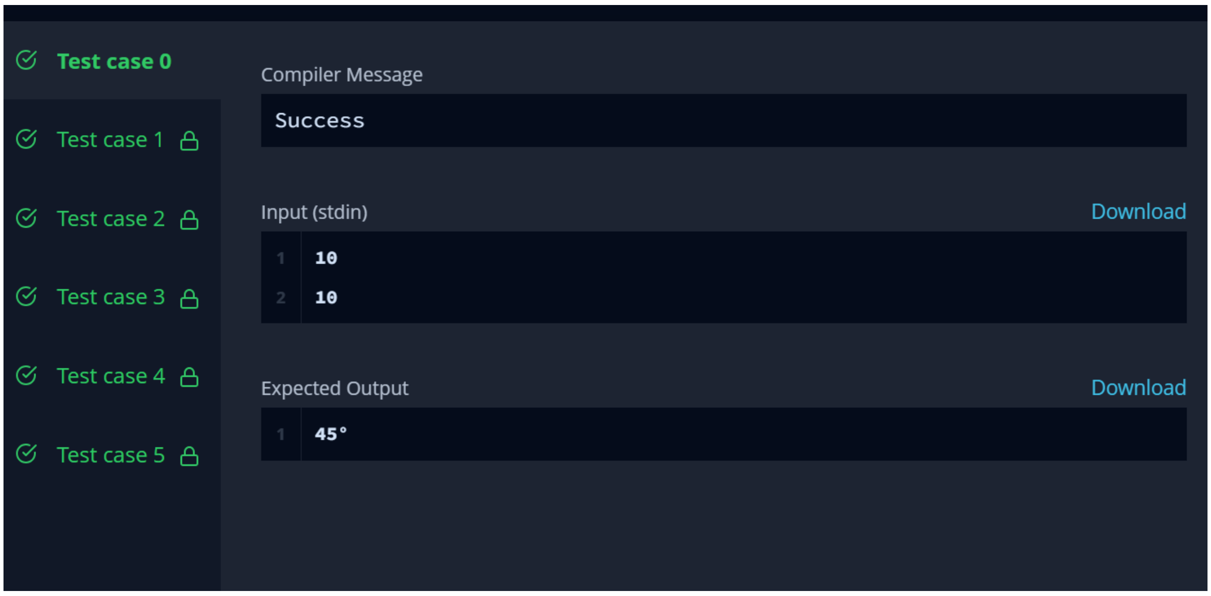
5. Find Angle MBC

Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
import math
ab=int(input())
bc=int(input())
ca=math.hypot(ab,bc)
mc=ca/2
bca=math.asin(1*ab/ca)
bm=math.sqrt((bc**2+mc**2)-(2*bc*mc*math.cos(bca)))
mbc=math.asin(math.sin(bca)*mc/bm)
print(int(round(math.degrees(mbc),0)),'\u00B0',sep='')
```

Output:



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

Compiler Message

```
Success
```

Input (stdin) Download

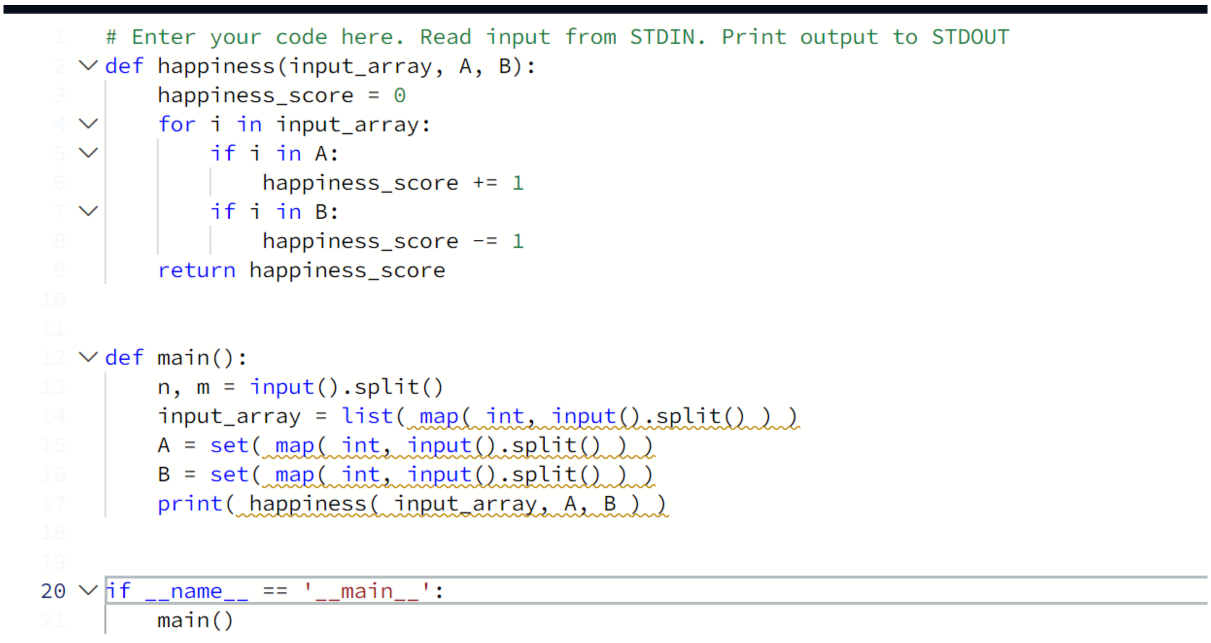
```
10
10
```

Expected Output Download

```
45°
```

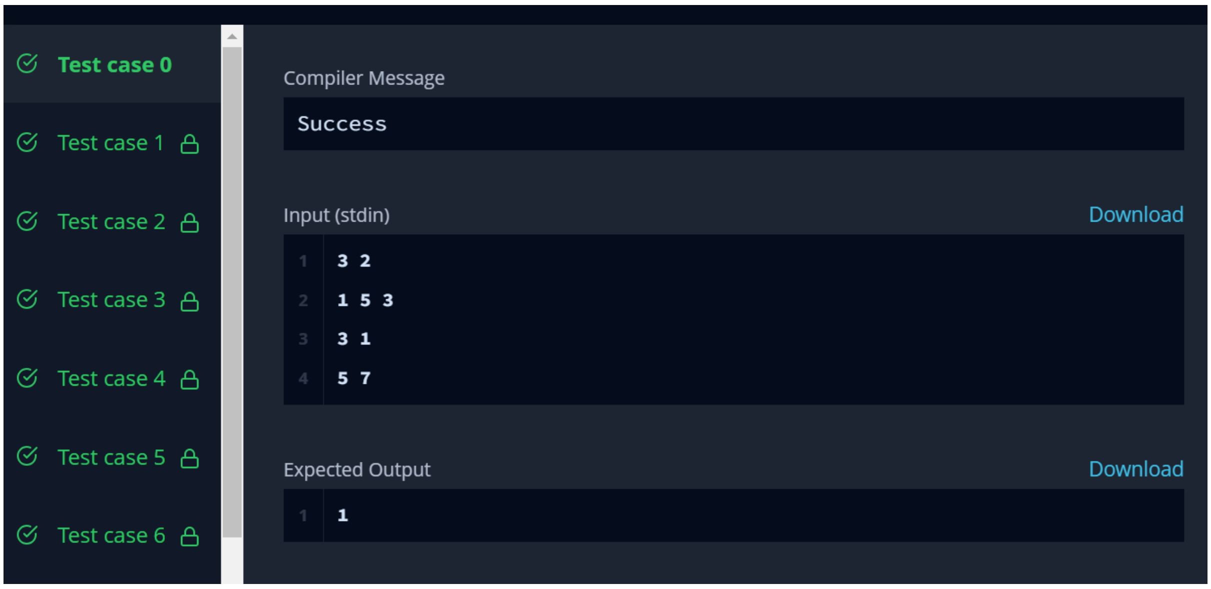
6. No Idea!

Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
def happiness(input_array, A, B):
    happiness_score = 0
    for i in input_array:
        if i in A:
            happiness_score += 1
        if i in B:
            happiness_score -= 1
    return happiness_score


def main():
    n, m = input().split()
    input_array = list( map( int, input().split() ) )
    A = set( map( int, input().split() ) )
    B = set( map( int, input().split() ) )
    print( happiness( input_array, A, B ) )


if __name__ == '__main__':
    main()
```

Output:



Compiler Message

```
Success
```

Input (stdin)

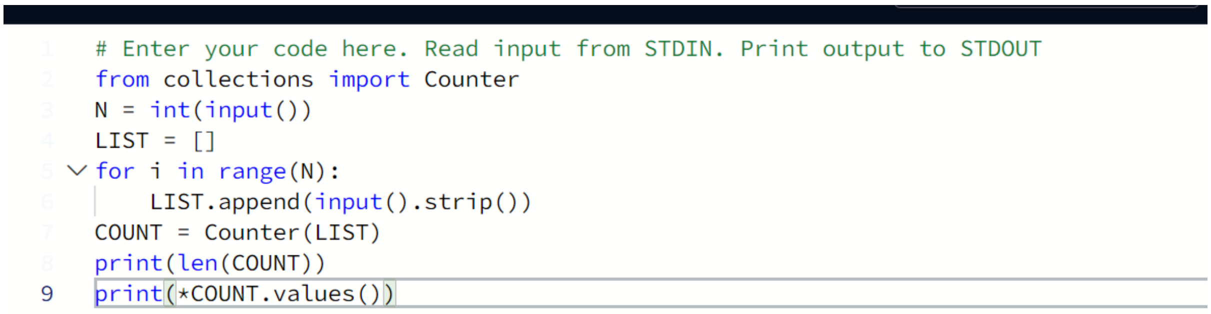
```
3 2
1 5 3
3 1
5 7
```

Expected Output

```
1
```

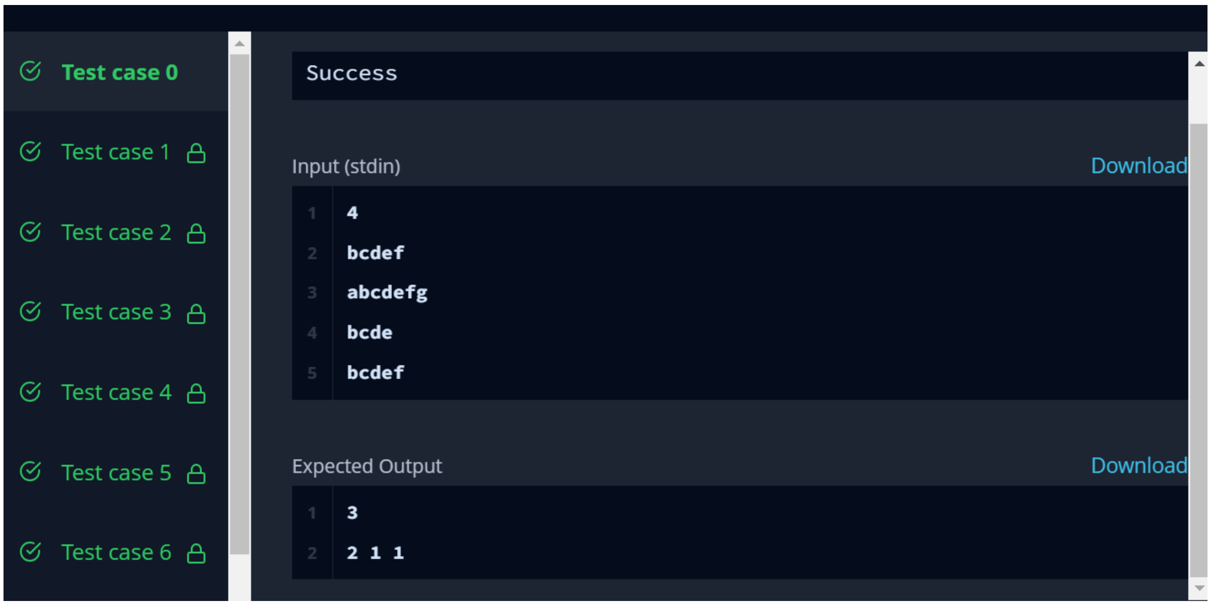
7. Word Order

Code:



```
# Enter your code here. Read input from STDIN. Print output to STDOUT
from collections import Counter
N = int(input())
LIST = []
for i in range(N):
    LIST.append(input().strip())
COUNT = Counter(LIST)
print(len(COUNT))
print(*COUNT.values())
```

Output:



Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

Success

Input (stdin) Download

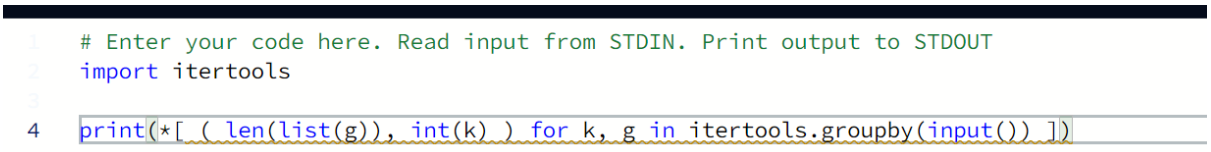
```
4
bcdef
abcdefg
bcde
bcdef
```

Expected Output Download

```
3
2 1 1
```

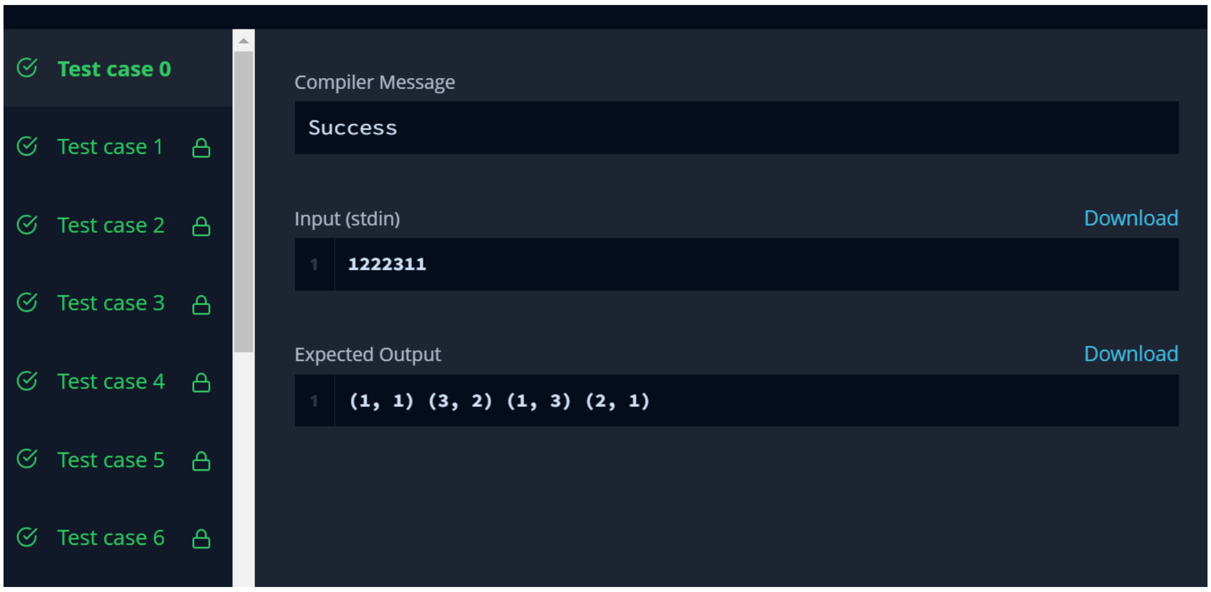
8. Compress the String!

Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
import itertools

print(*[ ( len(list(g)), int(k) ) for k, g in itertools.groupby(input()) ])
```

Output:



Compiler Message

```
Success
```

Input (stdin)

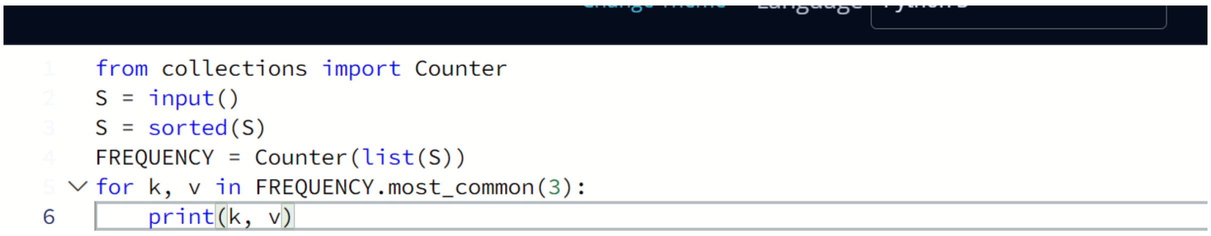
```
1222311
```

Expected Output

```
(1, 1) (3, 2) (1, 3) (2, 1)
```

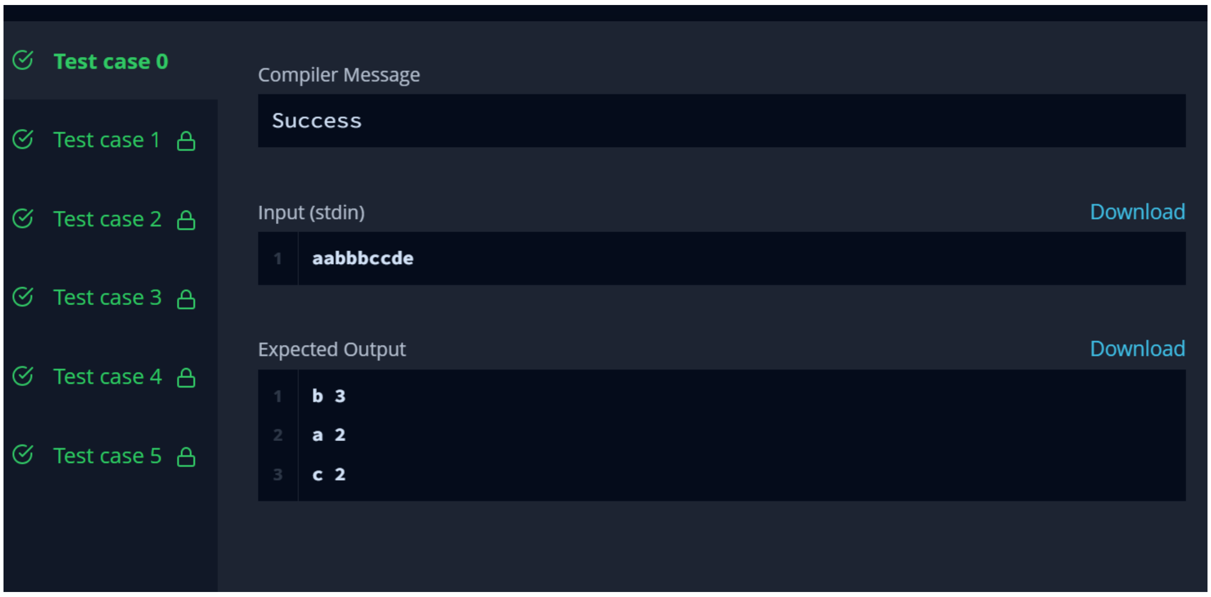
9. Company Logo

Code:



```python
from collections import Counter
S = input()
S = sorted(S)
FREQUENCY = Counter(list(S))
for k, v in FREQUENCY.most_common(3):
    print(k, v)
```

Output:



Compiler Message

```
Success
```

Input (stdin)

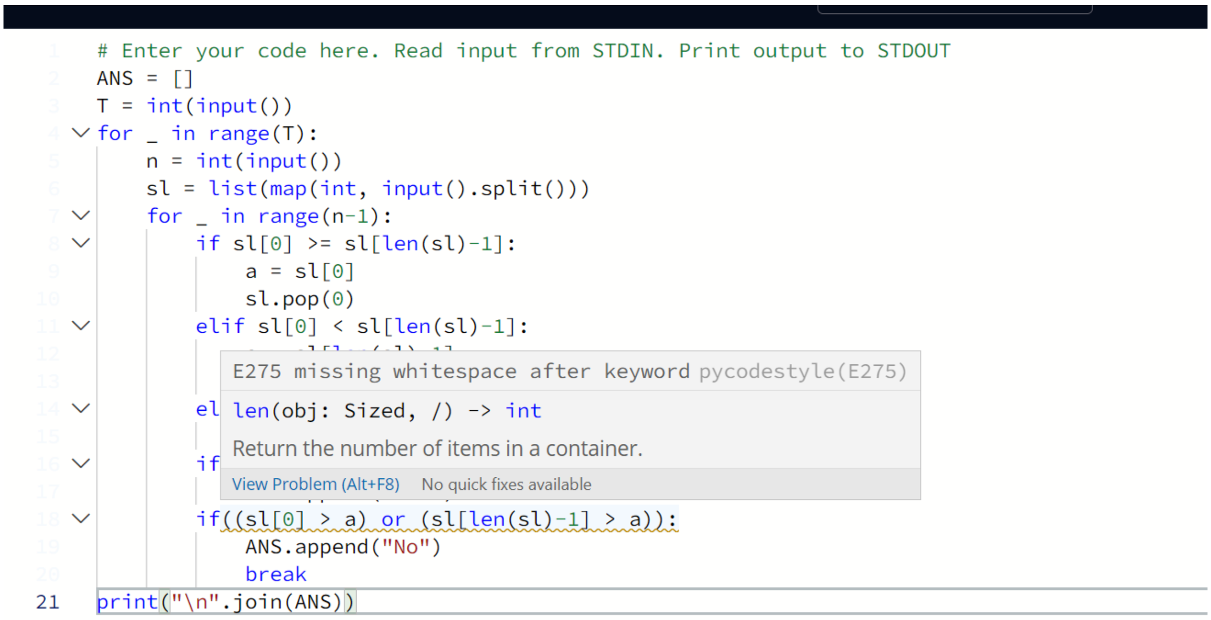
```
aabbbccde
```

Expected Output

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

10. Piling Up!

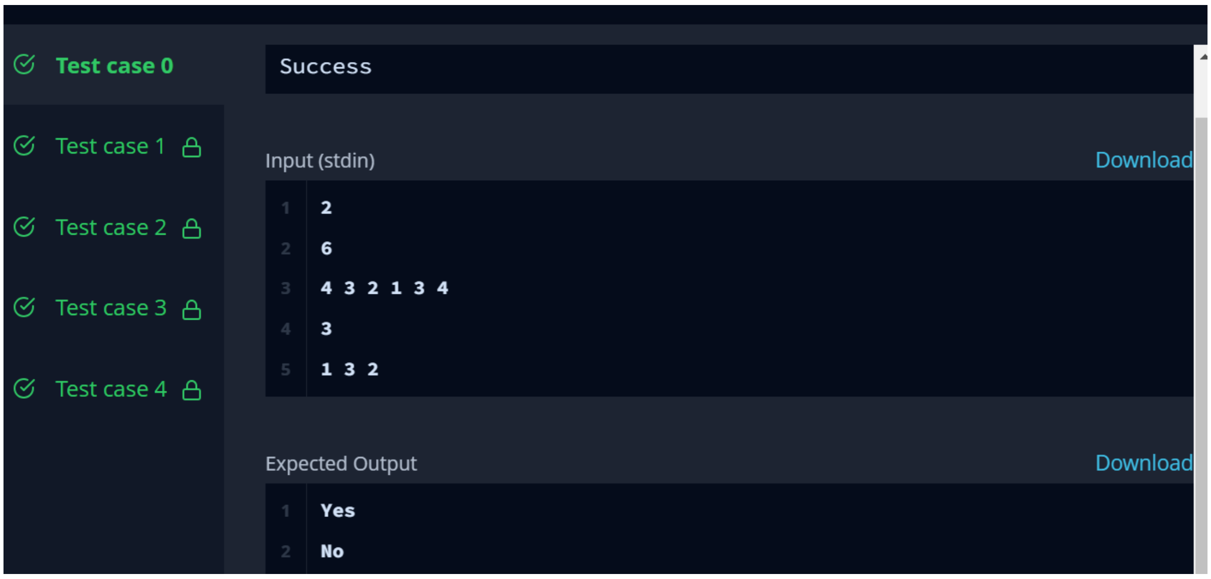
Code:



```
# Enter your code here. Read input from STDIN. Print output to STDOUT
ANS = []
T = int(input())
for _ in range(T):
    n = int(input())
    sl = list(map(int, input().split()))
    for _ in range(n-1):
        if sl[0] >= sl[len(sl)-1]:
            a = sl[0]
            sl.pop(0)
        elif sl[0] < sl[len(sl)-1]:

        el

        if

        if((sl[0] > a) or (sl[len(sl)-1] > a)):
            ANS.append("No")
            break
print("\n".join(ANS))
```

```
E275 missing whitespace after keyword pycodestyle(E275)
len(obj: Sized, /) -> int
Return the number of items in a container.
View Problem (Alt+F8)    No quick fixes available
```

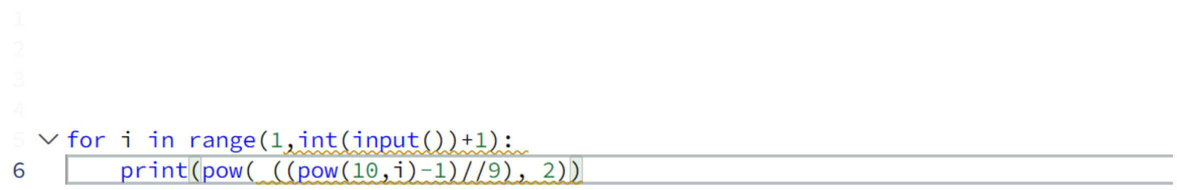
Output:



```
Success

Input (stdin)
2
6
4 3 2 1 3 4
3
1 3 2

Expected Output
Yes
No
```

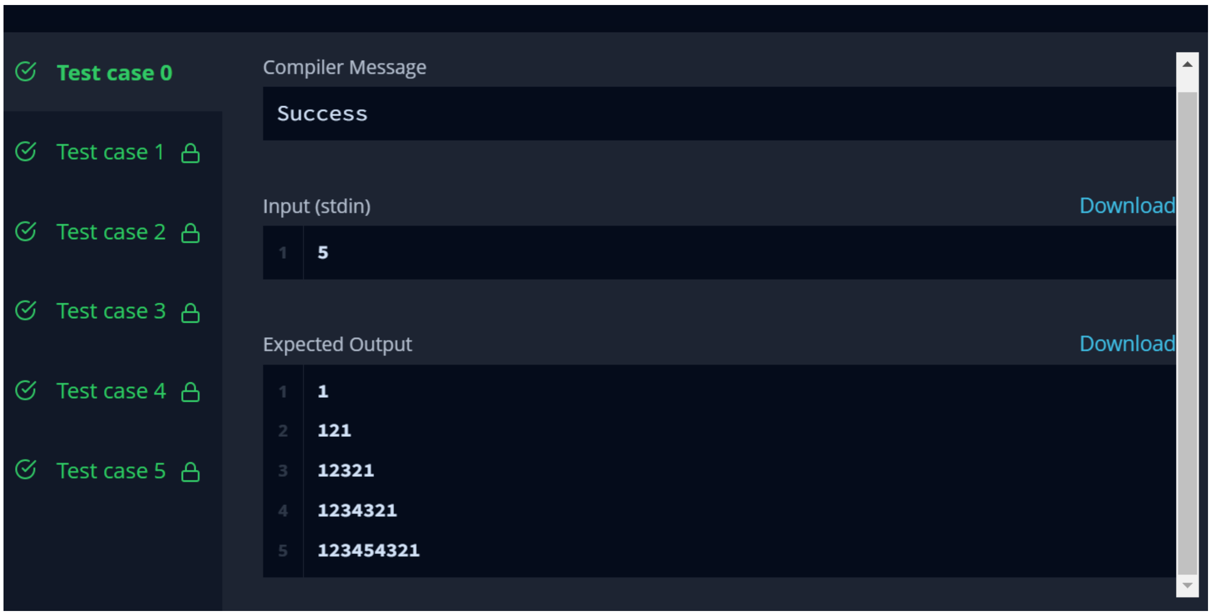
11. Triangle Quest 2

Code:



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

Output:



Compiler Message

```
Success
```

Input (stdin)

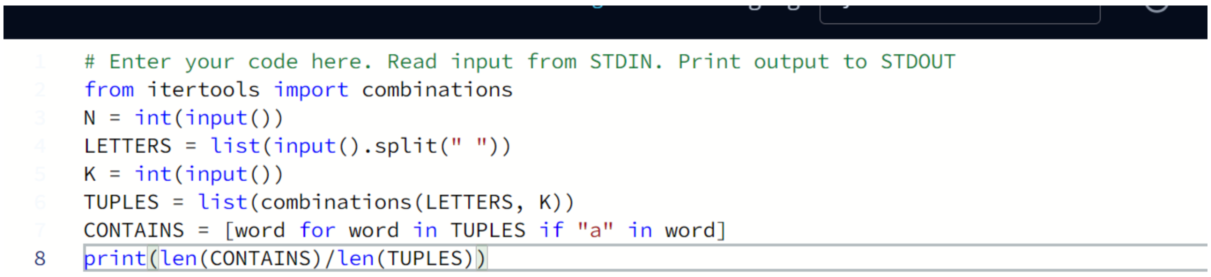
```
5
```

Expected Output

```
1
121
12321
1234321
123454321
```

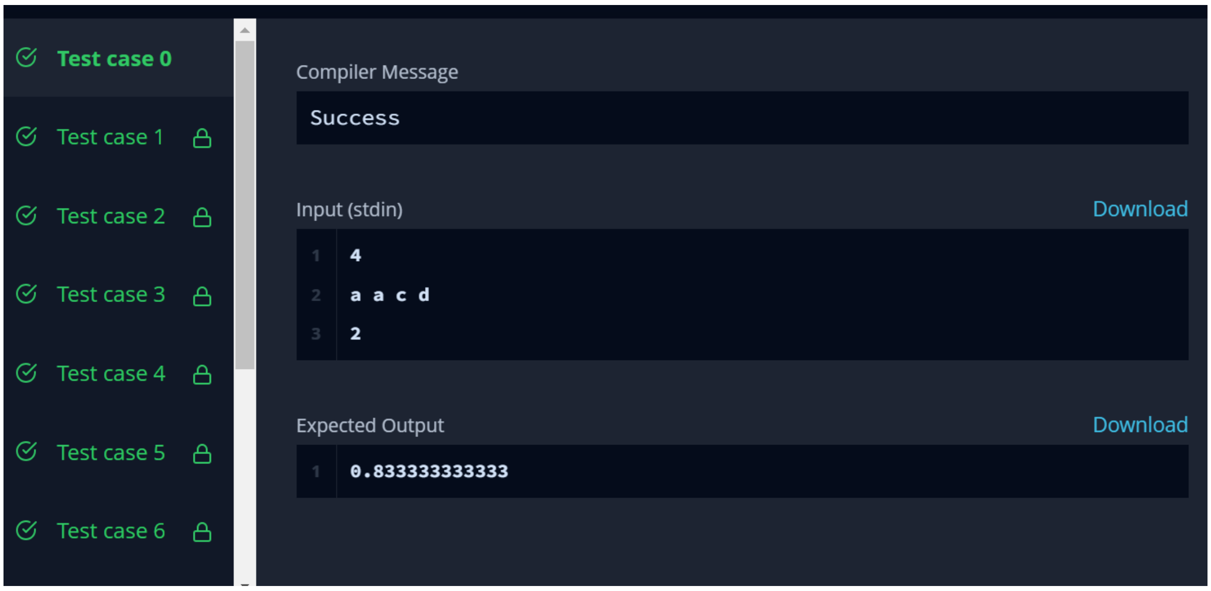
12. Iterables and Iterators

Code:



```
# Enter your code here. Read input from STDIN. Print output to STDOUT
from itertools import combinations
N = int(input())
LETTERS = list(input().split(" "))
K = int(input())
TUPLES = list(combinations(LETTERS, K))
CONTAINS = [word for word in TUPLES if "a" in word]
print(len(CONTAINS)/len(TUPLES))
```

Output:



Compiler Message

```
Success
```

Input (stdin)

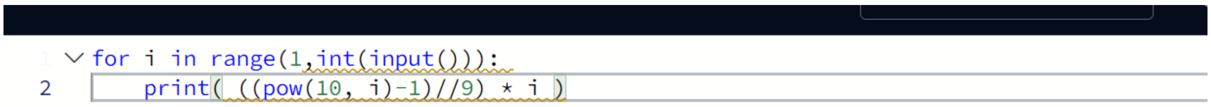
```
4
a a c d
2
```

Expected Output

```
0.833333333333
```

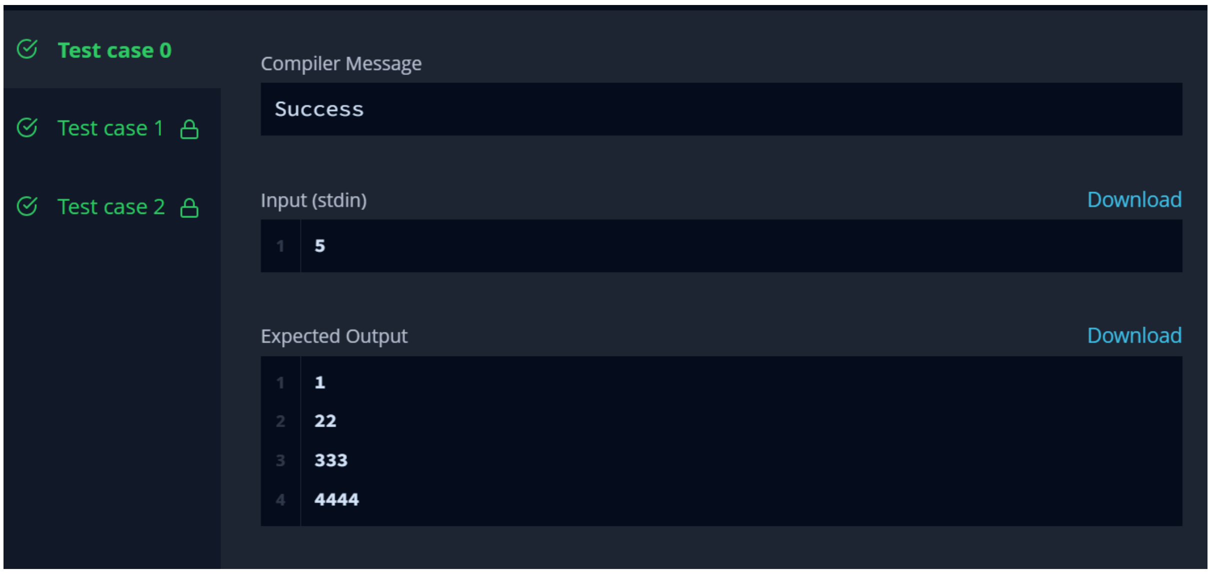
13. Triangle Quest

Code:



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

Output:



Compiler Message

```
Success
```

Input (stdin)

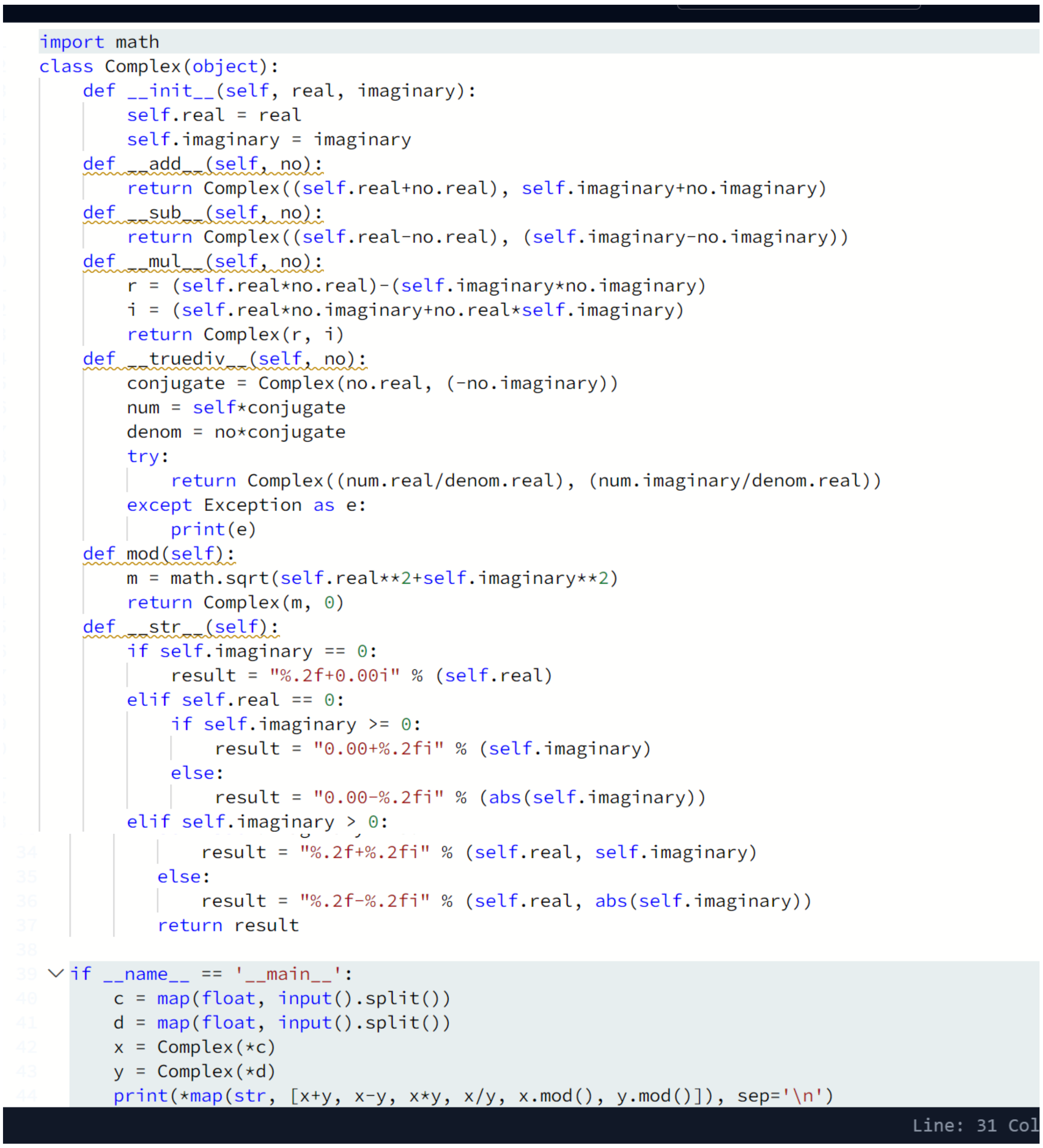
```
5
```

Expected Output

```
1
22
333
4444
```

14. Classes: Dealing with Complex Numbers

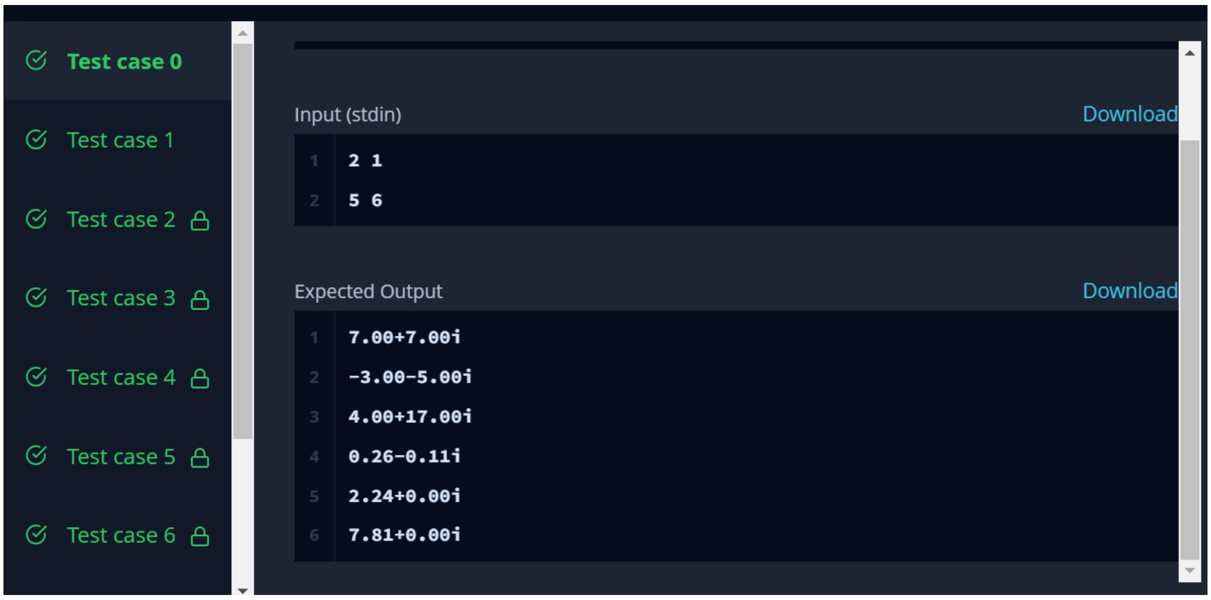
Code:



```
import math
class Complex(object):
    def __init__(self, real, imaginary):
        self.real = real
        self.imaginary = imaginary
    def __add__(self, no):
        return Complex((self.real+no.real), self.imaginary+no.imaginary)
    def __sub__(self, no):
        return Complex((self.real-no.real), (self.imaginary-no.imaginary))
    def __mul__(self, no):
        r = (self.real*no.real)-(self.imaginary*no.imaginary)
        i = (self.real*no.imaginary+no.real*self.imaginary)
        return Complex(r, i)
    def __truediv__(self, no):
        conjugate = Complex(no.real, (-no.imaginary))
        num = self*conjugate
        denom = no*conjugate
        try:
            return Complex((num.real/denom.real), (num.imaginary/denom.real))
        except Exception as e:
            print(e)
    def mod(self):
        m = math.sqrt(self.real**2+self.imaginary**2)
        return Complex(m, 0)
    def __str__(self):
        if self.imaginary == 0:
            result = "%.2f+0.00i" % (self.real)
        elif self.real == 0:
            if self.imaginary >= 0:
                result = "0.00+%.2fi" % (self.imaginary)
            else:
                result = "0.00-%.2fi" % (abs(self.imaginary))
        elif self.imaginary > 0:
            result = "%.2f+%.2fi" % (self.real, self.imaginary)
        else:
            result = "%.2f-%.2fi" % (self.real, abs(self.imaginary))
        return result

if __name__ == '__main__':
    c = map(float, input().split())
    d = map(float, input().split())
    x = Complex(*c)
    y = Complex(*d)
    print(*map(str, [x+y, x-y, x*y, x/y, x.mod(), y.mod()]), sep='\n')
```

Line: 31 Col

Output:

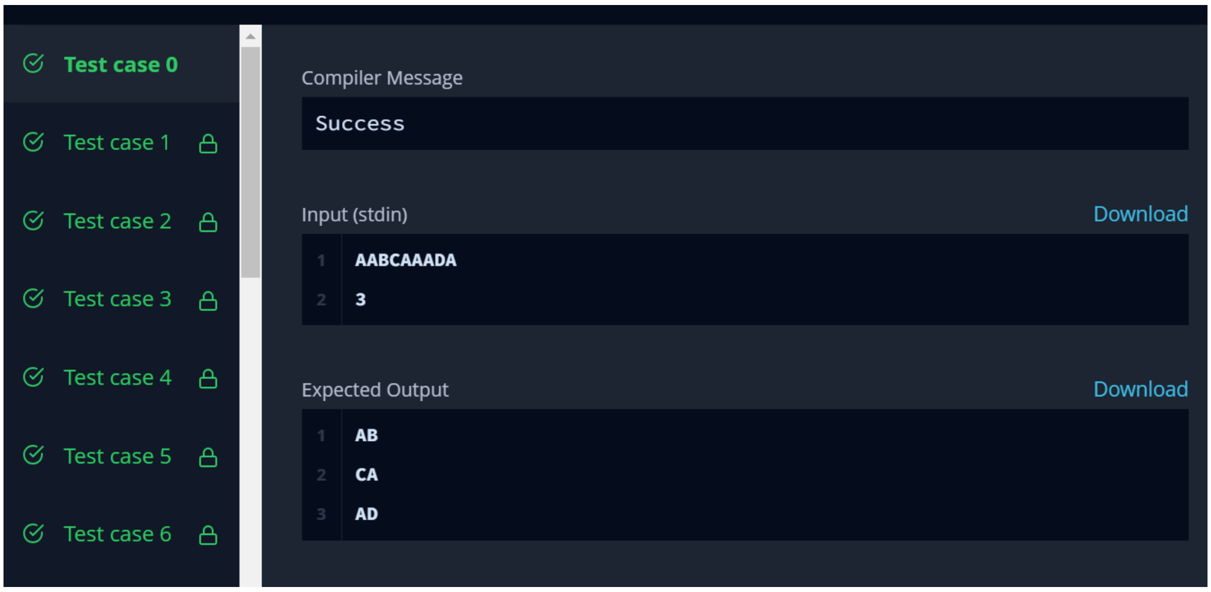


Input (stdin)

```
2 1
5 6
```

Expected Output

```
7.00+7.00i
-3.00-5.00i
4.00+17.00i
0.26-0.11i
2.24+0.00i
7.81+0.00i
```



Compiler Message

```
Success
```

Input (stdin)

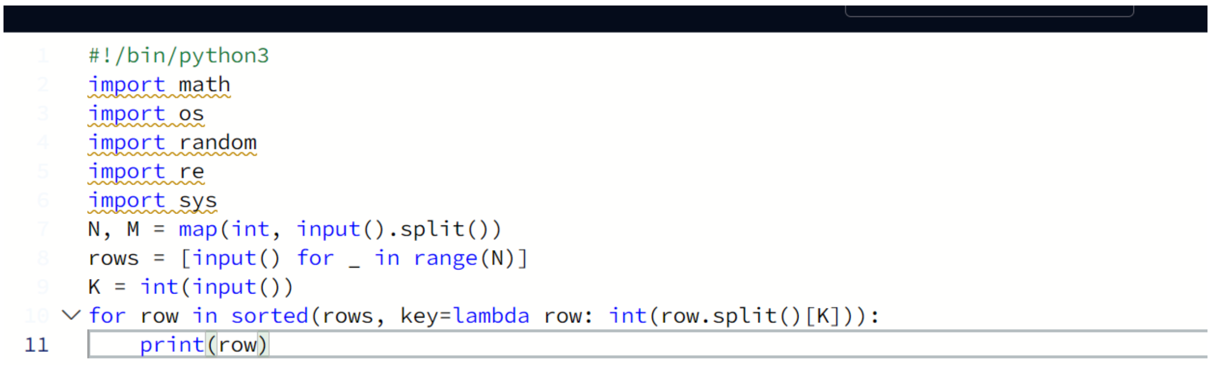
```
AABCAAADA
3
```

Expected Output

```
AB
CA
AD
```

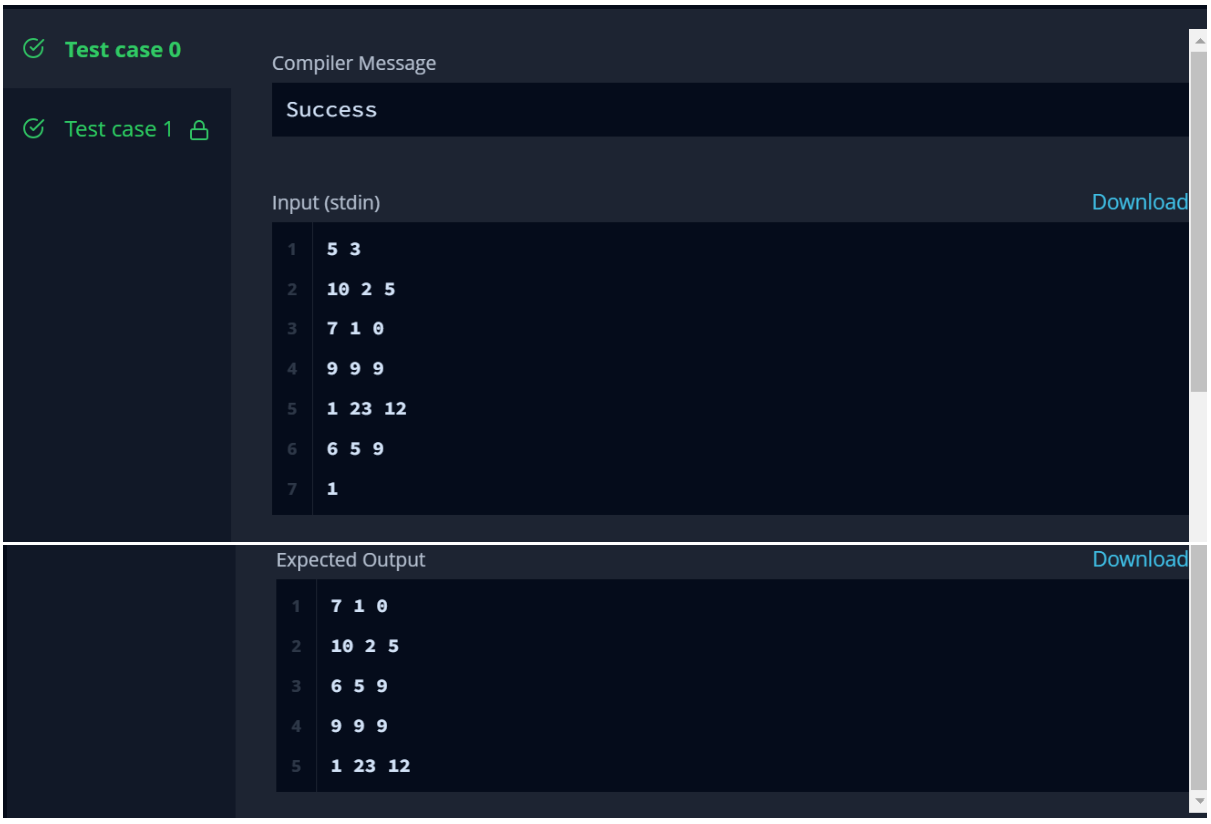
15. Athlete Sort

Code:



```python
#!/bin/python3
import math
import os
import random
import re
import sys
N, M = map(int, input().split())
rows = [input() for _ in range(N)]
K = int(input())
for row in sorted(rows, key=lambda row: int(row.split()[K])):
    print(row)
```

Output:



Compiler Message

```
Success
```

Input (stdin)

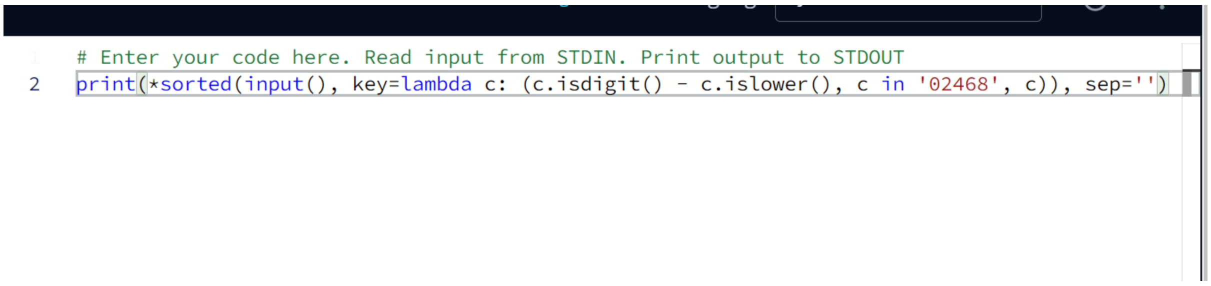
```
5 3
10 2 5
7 1 0
9 9 9
1 23 12
6 5 9
1
```

Expected Output

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

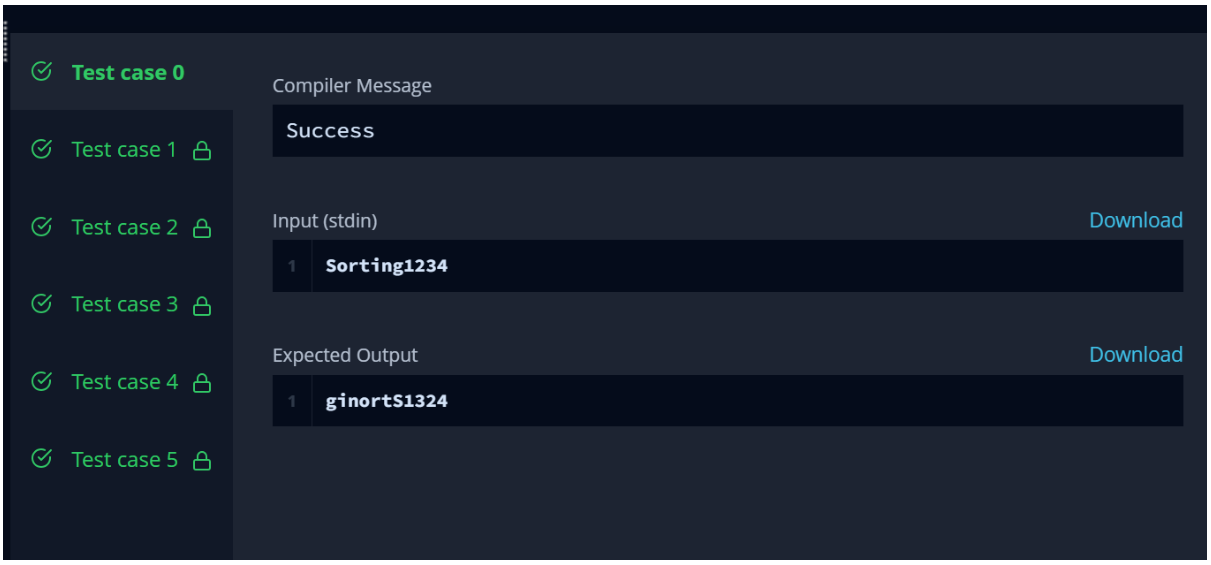
16. ginortS

Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
print(*sorted(input(), key=lambda c: (c.isdigit() - c.islower(), c in '02468', c)), sep='')
```

Output:



Compiler Message

```
Success
```

Input (stdin)

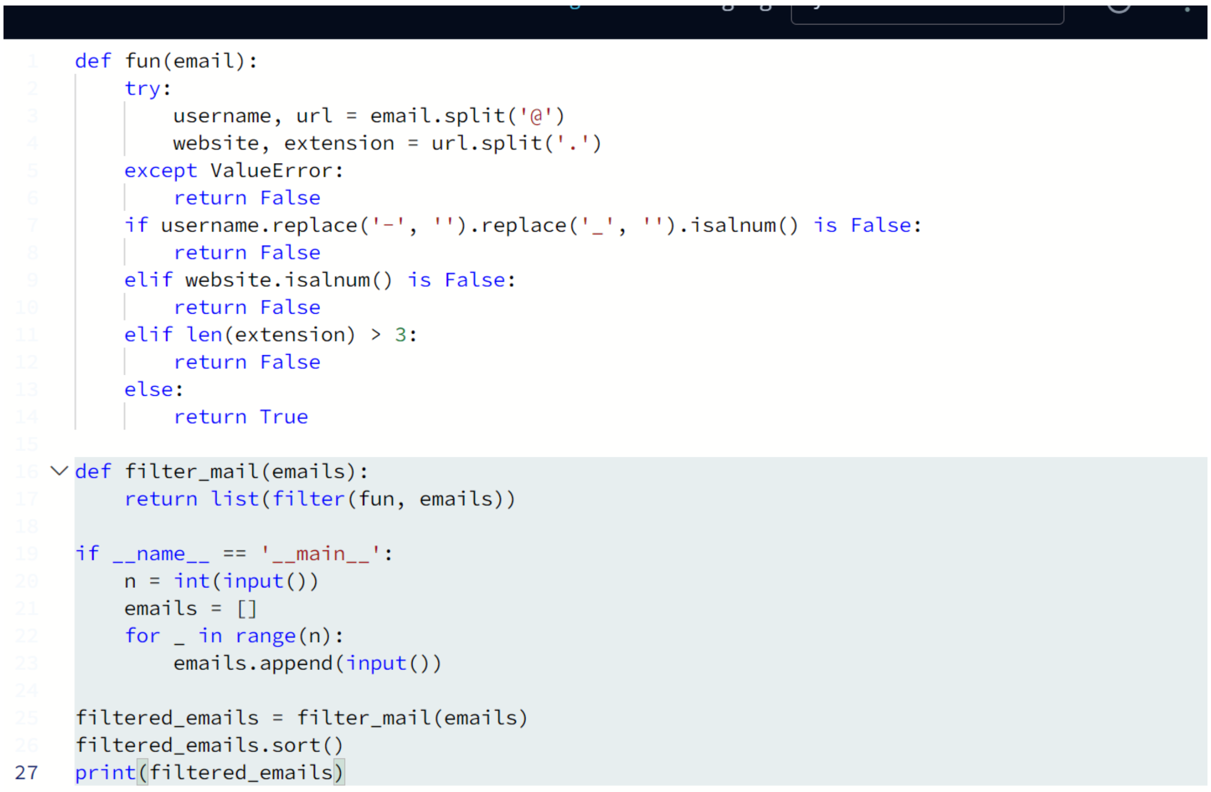
```
Sorting1234
```

Expected Output

```
ginortS1324
```

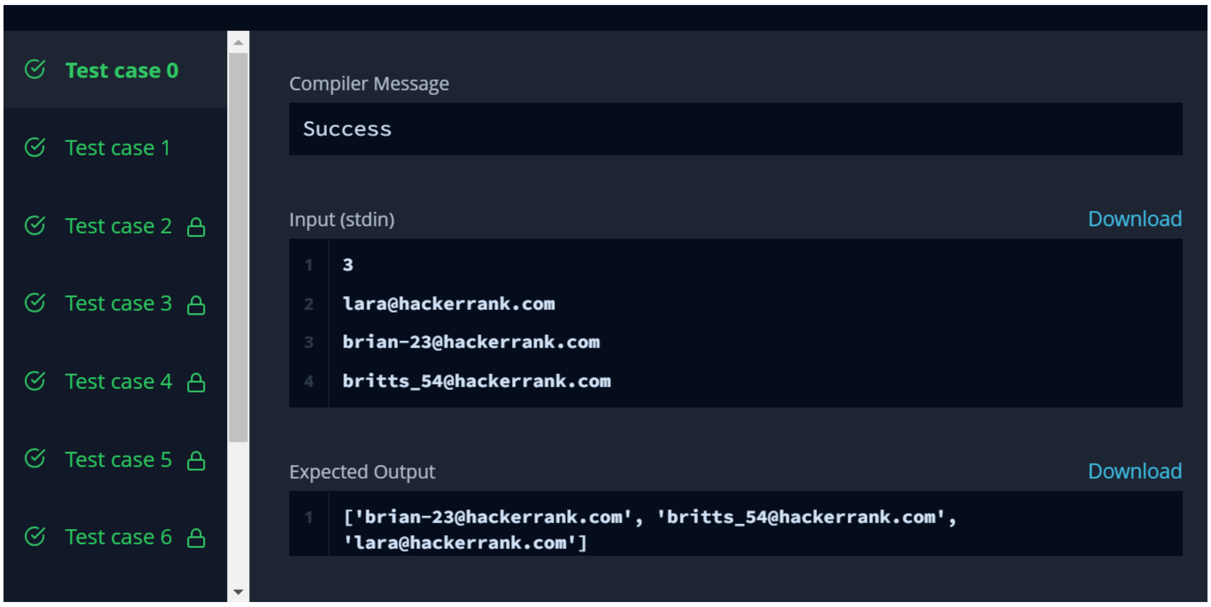
17. Validating Email Addresses With a Filter

Code:



```
def fun(email):
    try:
        username, url = email.split('@')
        website, extension = url.split('.')
    except ValueError:
        return False
    if username.replace('-', '').replace('_', '').isalnum() is False:
        return False
    elif website.isalnum() is False:
        return False
    elif len(extension) > 3:
        return False
    else:
        return True

def filter_mail(emails):
    return list(filter(fun, emails))

if __name__ == '__main__':
    n = int(input())
    emails = []
    for _ in range(n):
        emails.append(input())

filtered_emails = filter_mail(emails)
filtered_emails.sort()
print(filtered_emails)
```

Output:



Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

Compiler Message

```
Success
```

Input (stdin) Download

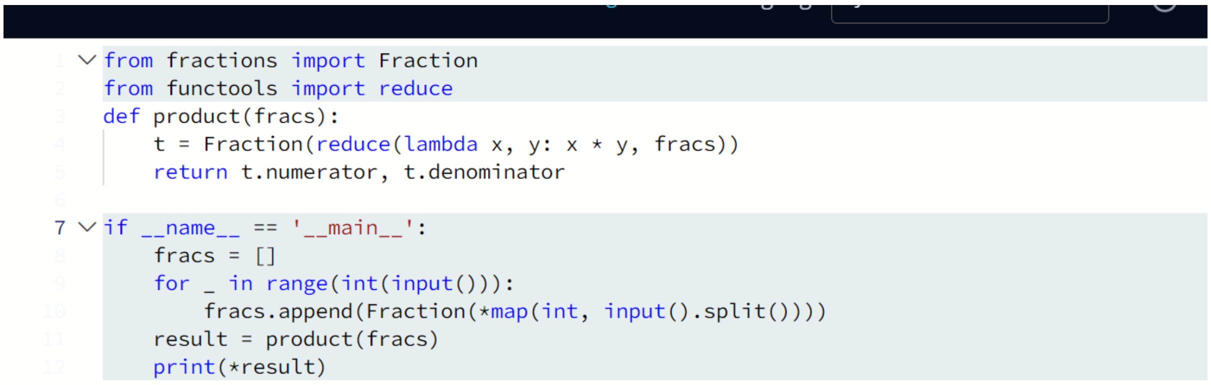
```
3
lara@hackerrank.com
brian-23@hackerrank.com
britts_54@hackerrank.com
```

Expected Output Download

```
['brian-23@hackerrank.com', 'britts_54@hackerrank.com',
'lara@hackerrank.com']
```

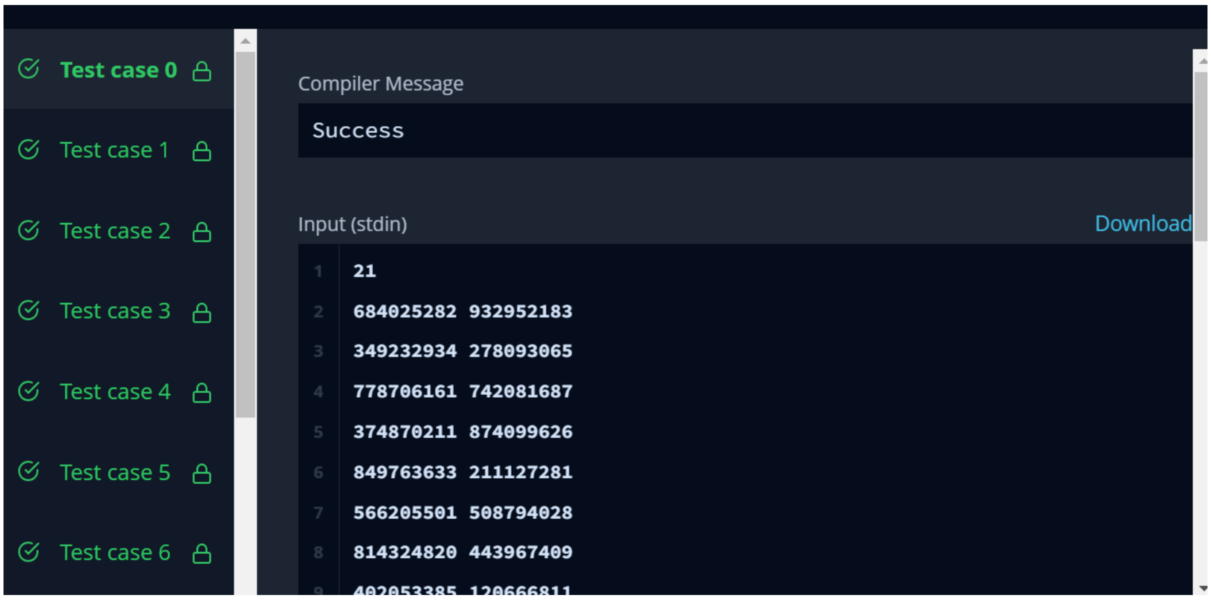
18. Reduce Function

Code:



```python
from fractions import Fraction
from functools import reduce
def product(fracs):
    t = Fraction(reduce(lambda x, y: x * y, fracs))
    return t.numerator, t.denominator

if __name__ == '__main__':
    fracs = []
    for _ in range(int(input())):
        fracs.append(Fraction(*map(int, input().split())))
    result = product(fracs)
    print(*result)
```

Output:



Compiler Message

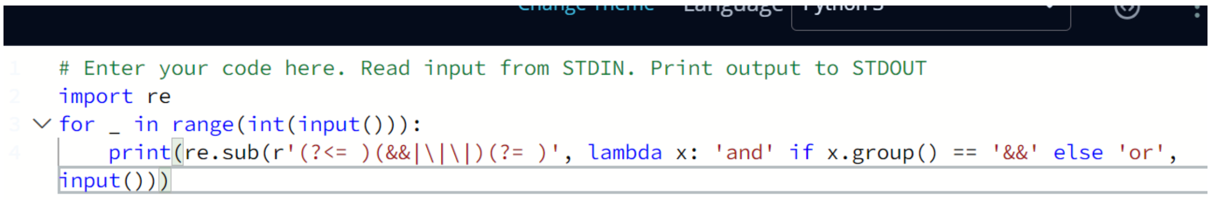
```
Success
```

Input (stdin)

```
21
684025282 932952183
349232934 278093065
778706161 742081687
374870211 874099626
849763633 211127281
566205501 508794028
814324820 443967409
402053385 120666811
```

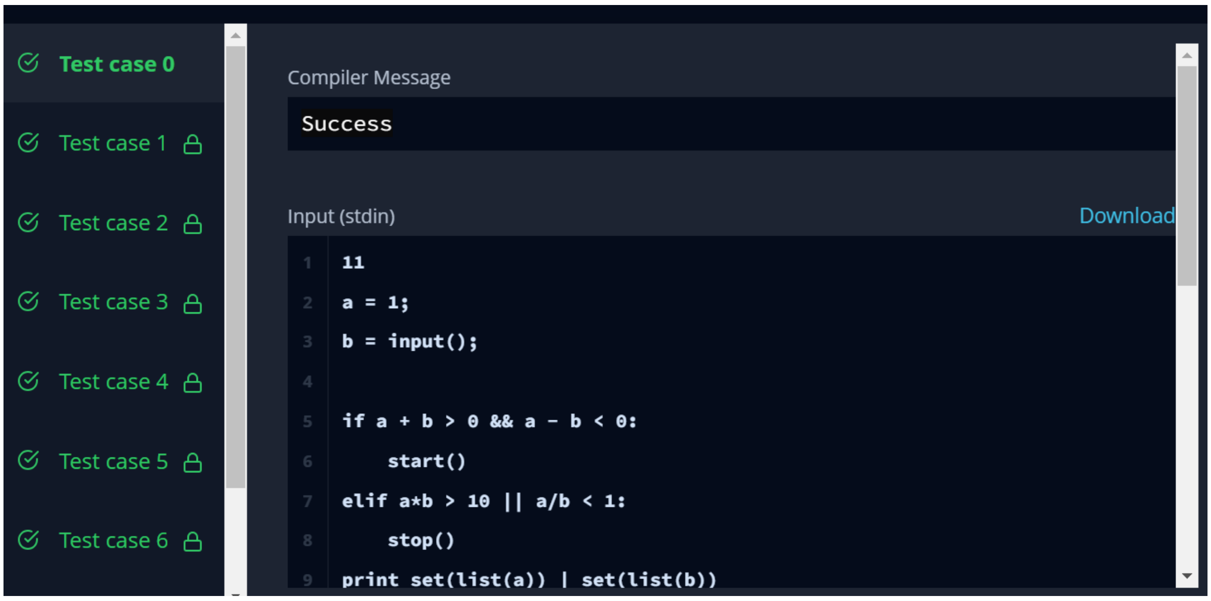
19. Regex Substitution

Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
import re
for _ in range(int(input())):
    print(re.sub(r'(?<= )(&&|\|\|)(?= )', lambda x: 'and' if x.group() == '&&' else 'or', input()))
```

Output:

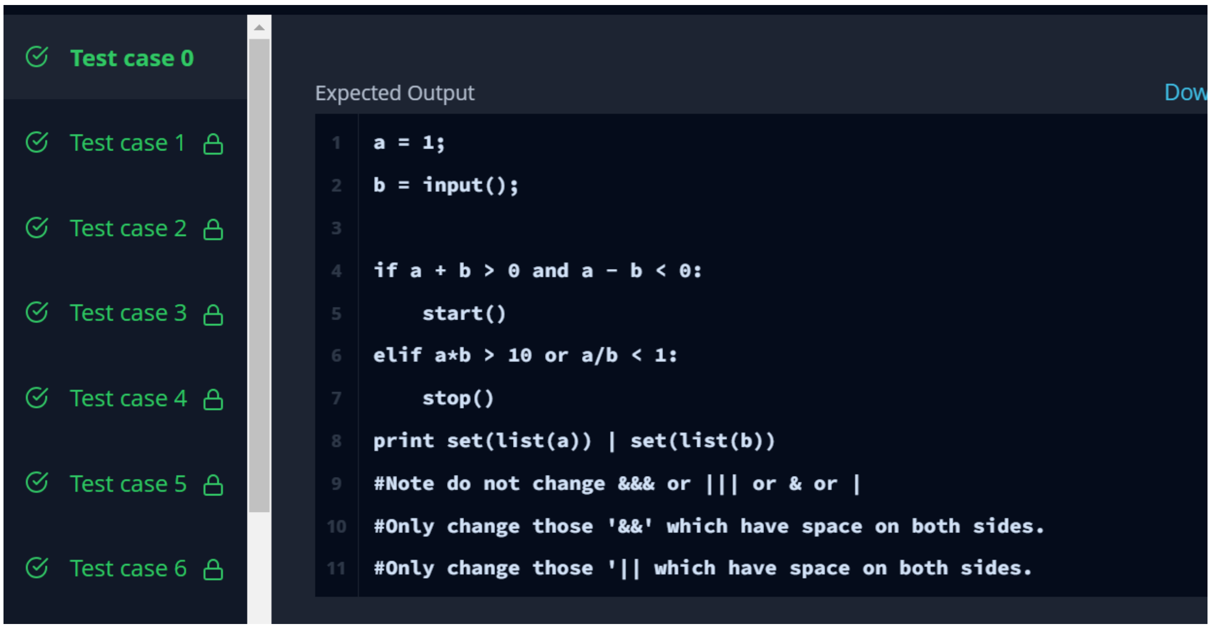


Compiler Message

```
Success
```

Input (stdin)

```
11
a = 1;
b = input();

if a + b > 0 && a - b < 0:
    start()
elif a*b > 10 || a/b < 1:
    stop()
print set(list(a)) | set(list(b))
```

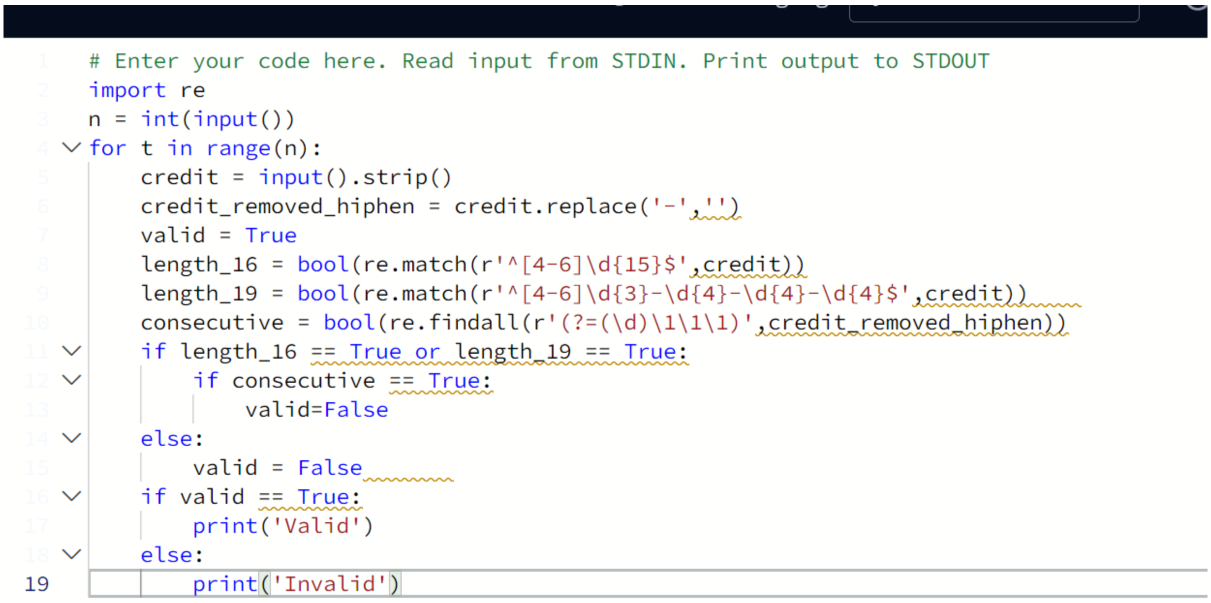


Expected Output

```
a = 1;
b = input();

if a + b > 0 and a - b < 0:
    start()
elif a*b > 10 or a/b < 1:
    stop()
print set(list(a)) | set(list(b))
#Note do not change &&& or ||| or & or |
#Only change those '&&' which have space on both sides.
#Only change those '|| which have space on both sides.
```

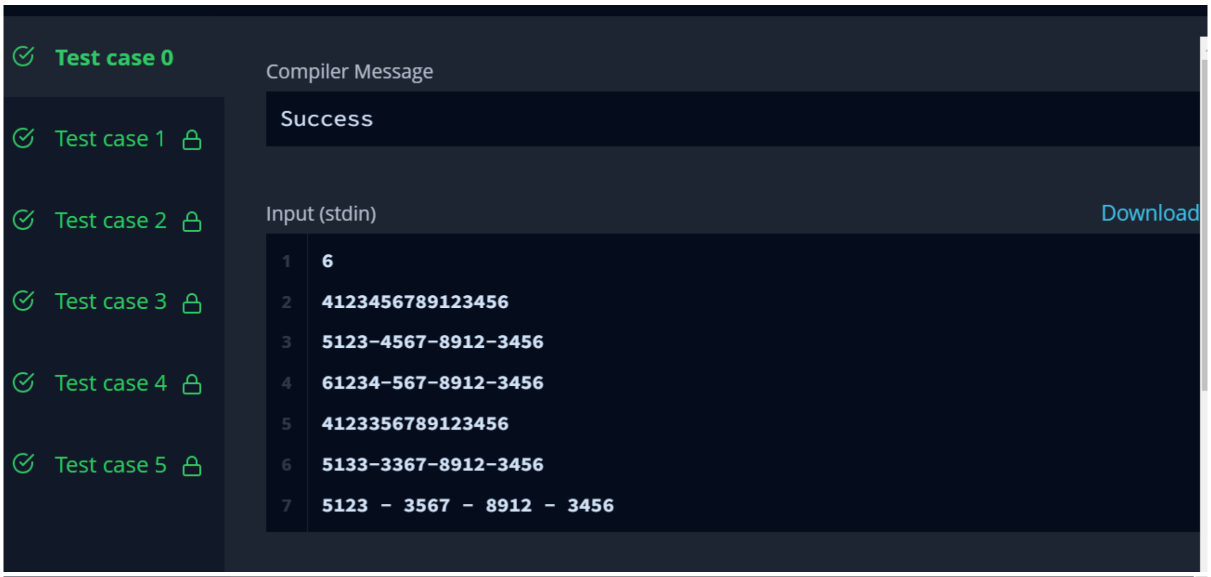
20. Validating Credit Card Numbers

Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
import re
n = int(input())
for t in range(n):
    credit = input().strip()
    credit_removed_hiphen = credit.replace('-','')
    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\1)',credit_removed_hiphen))
    if length_16 == True or length_19 == True:
        if consecutive == True:
            valid=False
    else:
        valid = False
    if valid == True:
        print('Valid')
    else:
        print('Invalid')
```

Output:



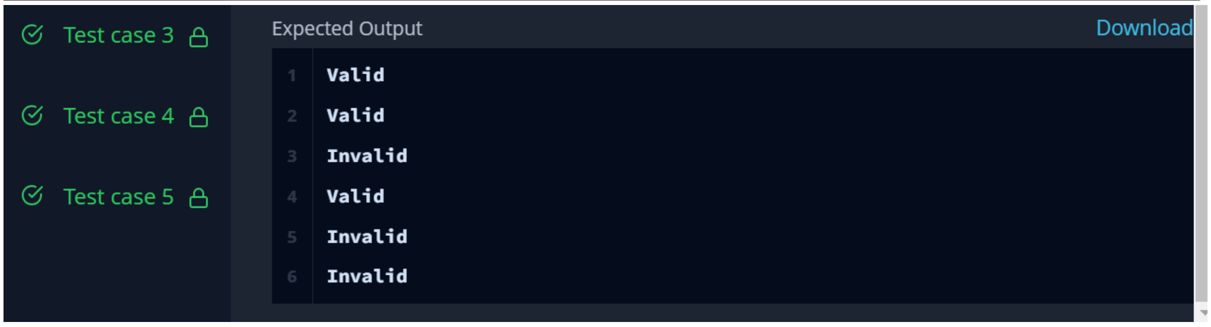
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5

Compiler Message

```
Success
```

Input (stdin) Download

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



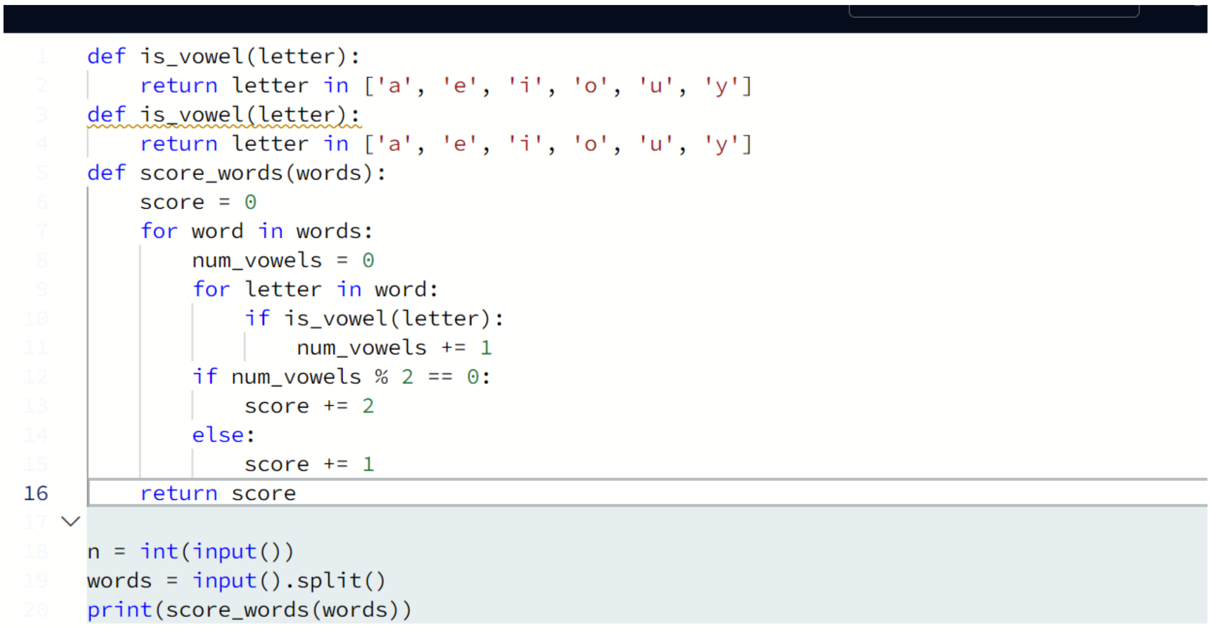
Test case 3
Test case 4
Test case 5

Expected Output Download

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

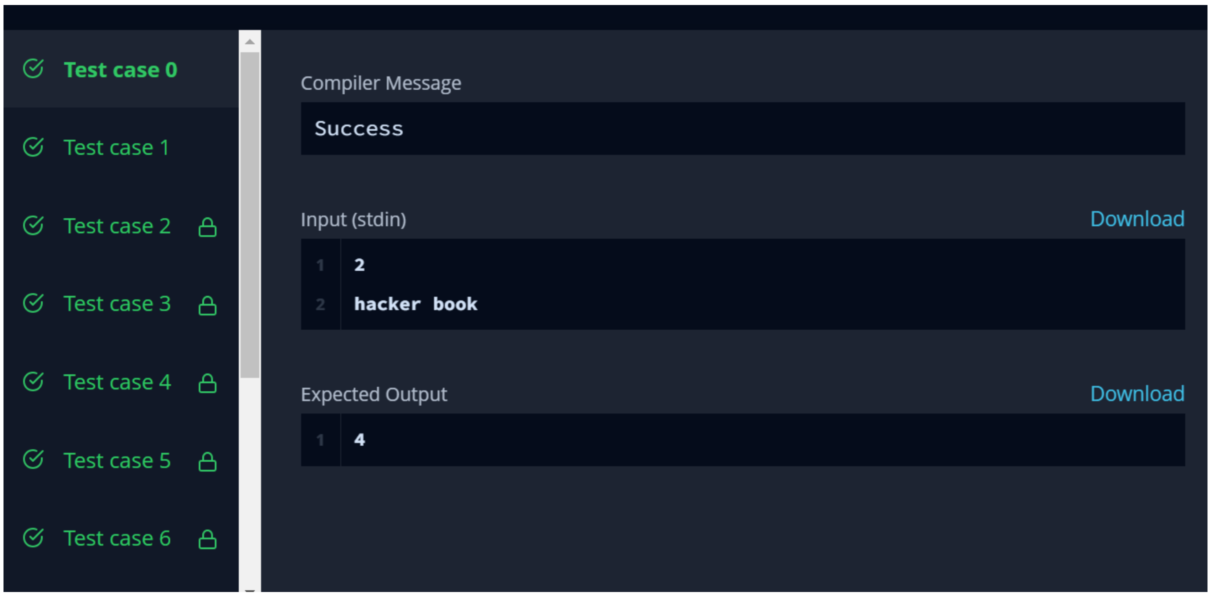
21. Words Score

Code:



```
def is_vowel(letter):
    return letter in ['a', 'e', 'i', 'o', 'u', 'y']
def is_vowel(letter):
    return letter in ['a', 'e', 'i', 'o', 'u', 'y']
def score_words(words):
    score = 0
    for word in words:
        num_vowels = 0
        for letter in word:
            if is_vowel(letter):
                num_vowels += 1
        if num_vowels % 2 == 0:
            score += 2
        else:
            score += 1
    return score

n = int(input())
words = input().split()
print(score_words(words))
```

Output:



Compiler Message

```
Success
```

Input (stdin)

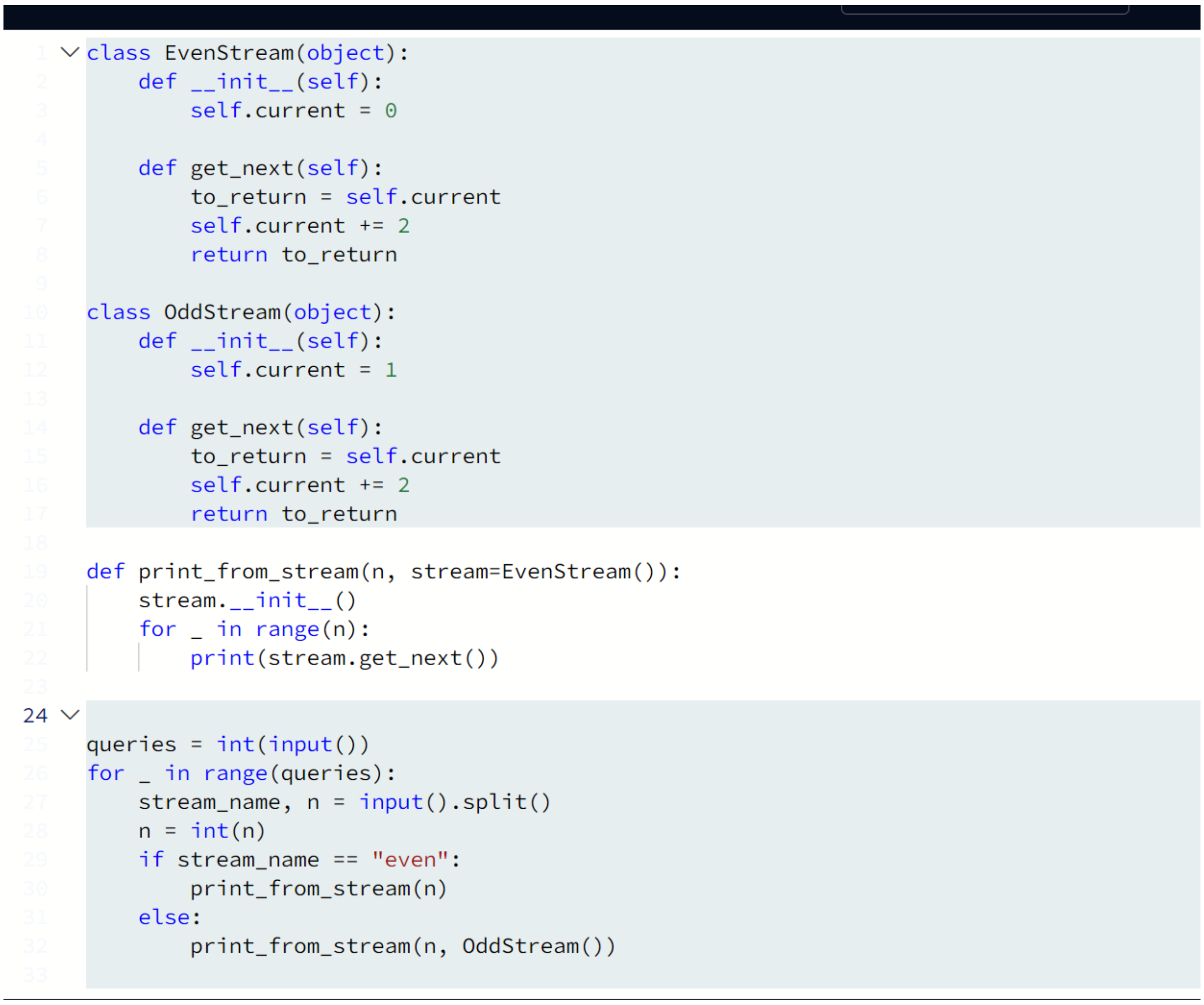
```
2
hacker book
```

Expected Output

```
4
```

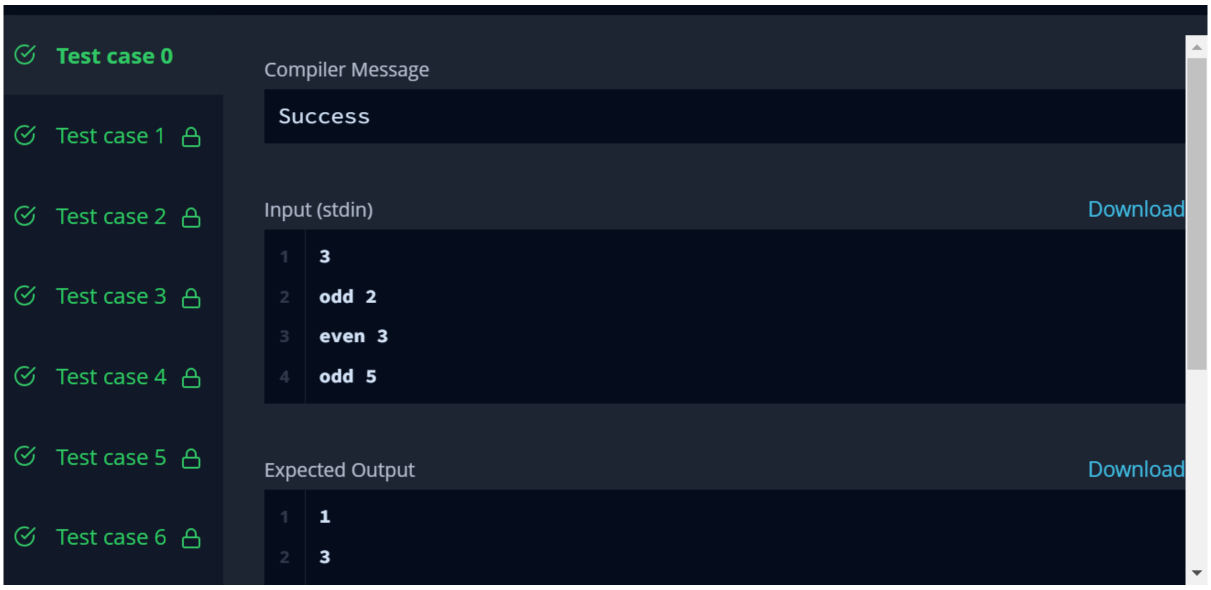
22. Default Arguments

Code:



```
class EvenStream(object):
    def __init__(self):
        self.current = 0

    def get_next(self):
        to_return = self.current
        self.current += 2
        return to_return

class OddStream(object):
    def __init__(self):
        self.current = 1

    def get_next(self):
        to_return = self.current
        self.current += 2
        return to_return

def print_from_stream(n, stream=EvenStream()):
    stream.__init__()
    for _ in range(n):
        print(stream.get_next())


queries = int(input())
for _ in range(queries):
    stream_name, n = input().split()
    n = int(n)
    if stream_name == "even":
        print_from_stream(n)
    else:
        print_from_stream(n, OddStream())
```

Output:



Compiler Message

```
Success
```

Input (stdin)

```
3
odd 2
even 3
odd 5
```

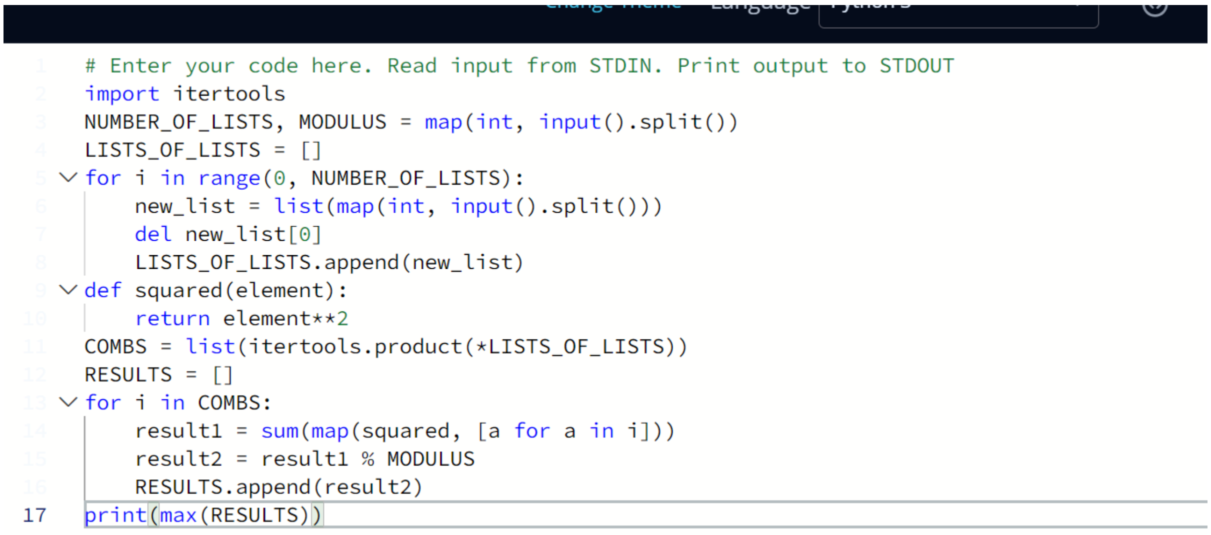
Expected Output

```
1
3
```

HARD DIFFICULTY CHALLENGES

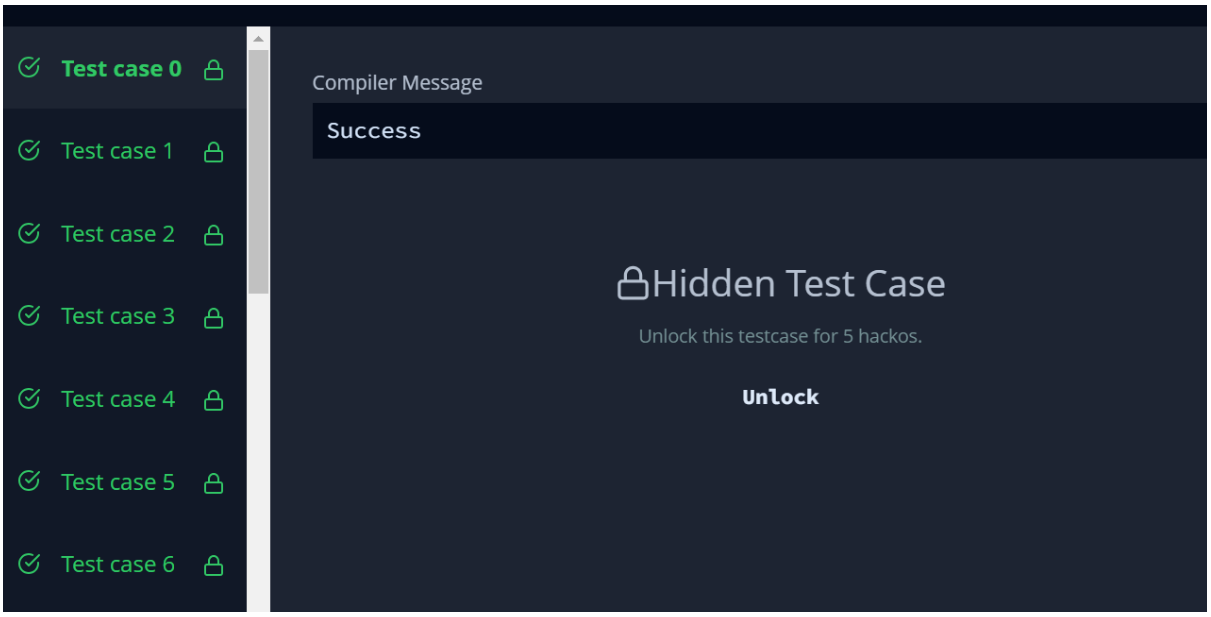
1. **Maximize It!**

   Code:



```python
# Enter your code here. Read input from STDIN. Print output to STDOUT
import itertools
NUMBER_OF_LISTS, MODULUS = map(int, input().split())
LISTS_OF_LISTS = []
for i in range(0, NUMBER_OF_LISTS):
    new_list = list(map(int, input().split()))
    del new_list[0]
    LISTS_OF_LISTS.append(new_list)
def squared(element):
    return element**2
COMBS = list(itertools.product(*LISTS_OF_LISTS))
RESULTS = []
for i in COMBS:
    result1 = sum(map(squared, [a for a in i]))
    result2 = result1 % MODULUS
    RESULTS.append(result2)
print(max(RESULTS))
```

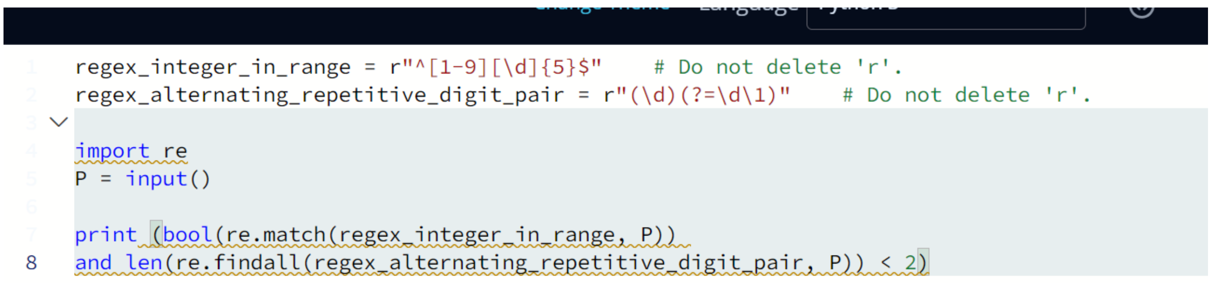
   Output:



```
Success
```

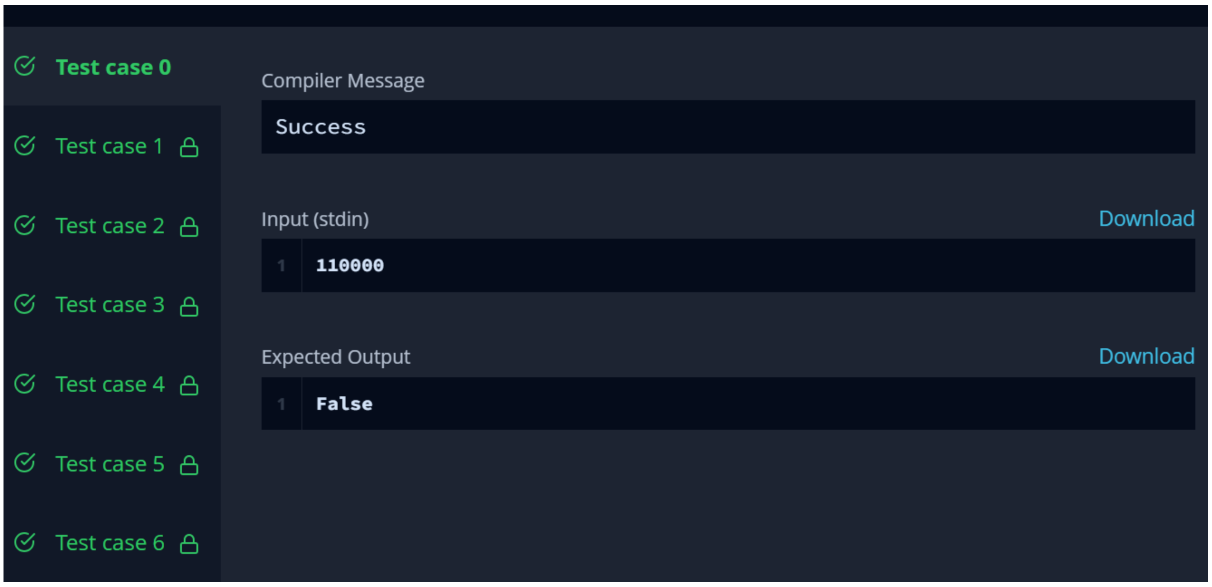
2. Validating Postal Codes

Code:



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

import re
P = input()

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

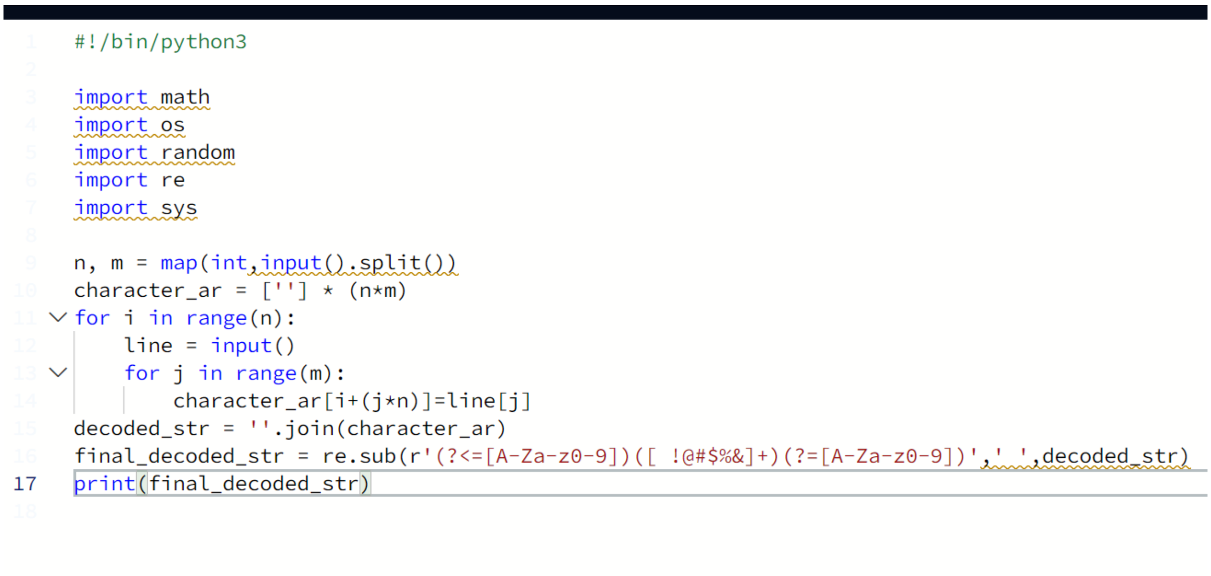
Output:



```
Compiler Message
Success

Input (stdin)
110000

Expected Output
False
```

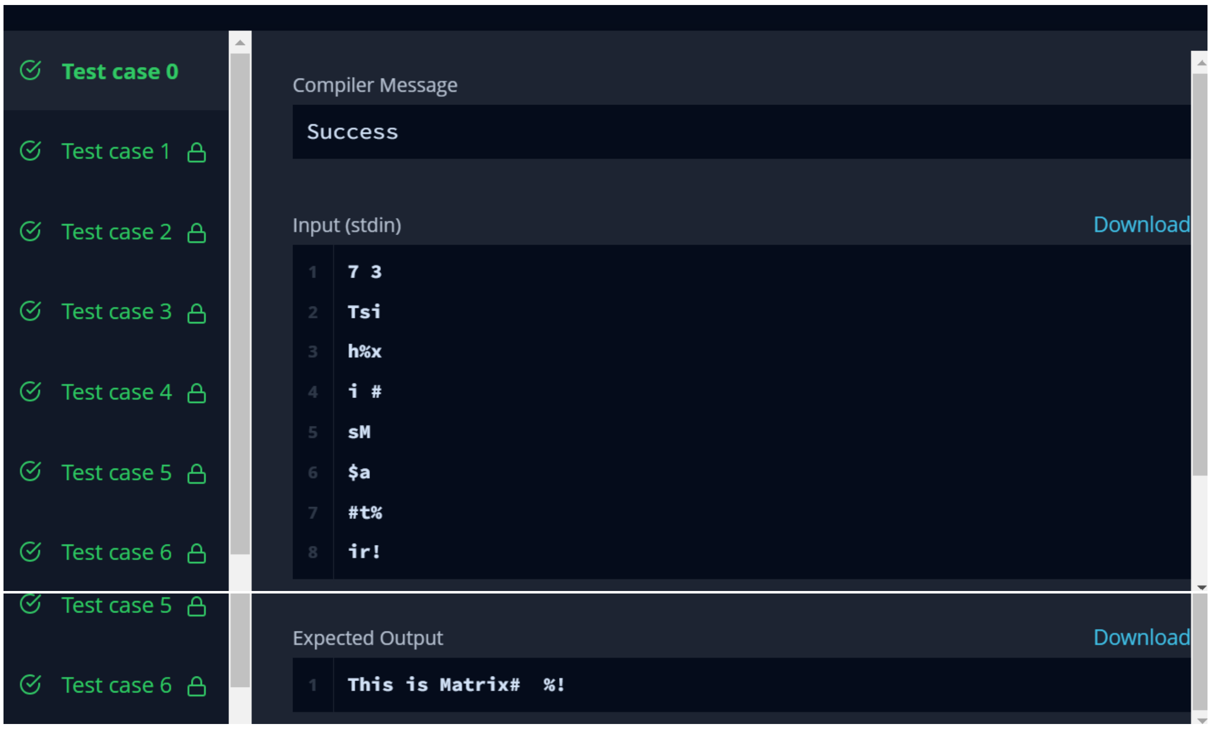
3. Matrix Script

Code:



```python
#!/bin/python3

import math
import os
import random
import re
import sys

n, m = map(int,input().split())
character_ar = [''] * (n*m)
for i in range(n):
    line = input()
    for j in range(m):
        character_ar[i+(j*n)]=line[j]
decoded_str = ''.join(character_ar)
final_decoded_str = re.sub(r'(?<=[A-Za-z0-9])([ !@#$%&]+)(?=[A-Za-z0-9])',' ',decoded_str)
print(final_decoded_str)
```

Output:



Compiler Message

```
Success
```

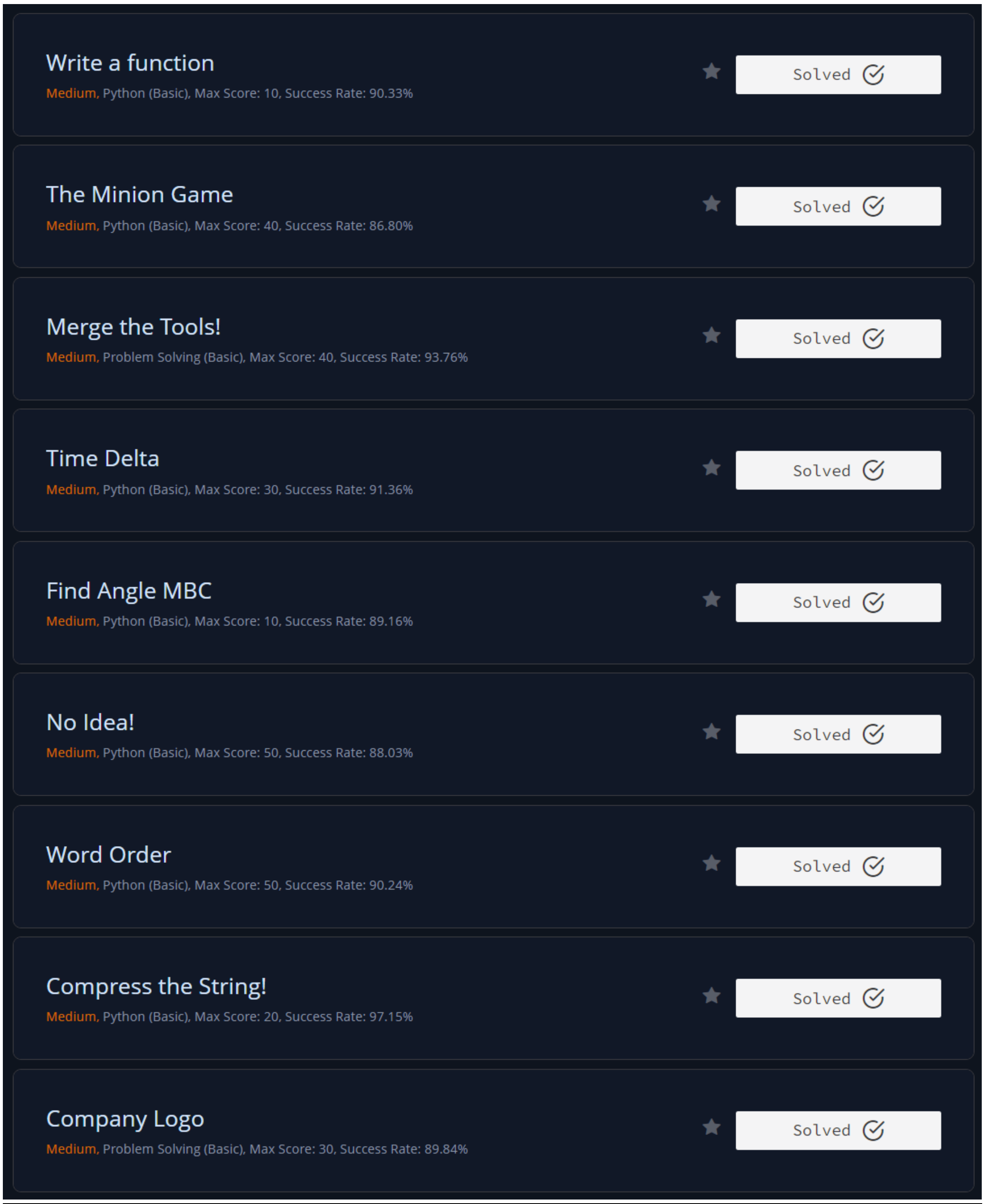
Input (stdin) Download

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

Expected Output Download

```
This is Matrix#  %!
```

ALL SOLVED CHALLENGES



Write a function
Medium, Python (Basic), Max Score: 10, Success Rate: 90.33%

The Minion Game
Medium, Python (Basic), Max Score: 40, Success Rate: 86.80%

Merge the Tools!
Medium, Problem Solving (Basic), Max Score: 40, Success Rate: 93.76%

Time Delta
Medium, Python (Basic), Max Score: 30, Success Rate: 91.36%

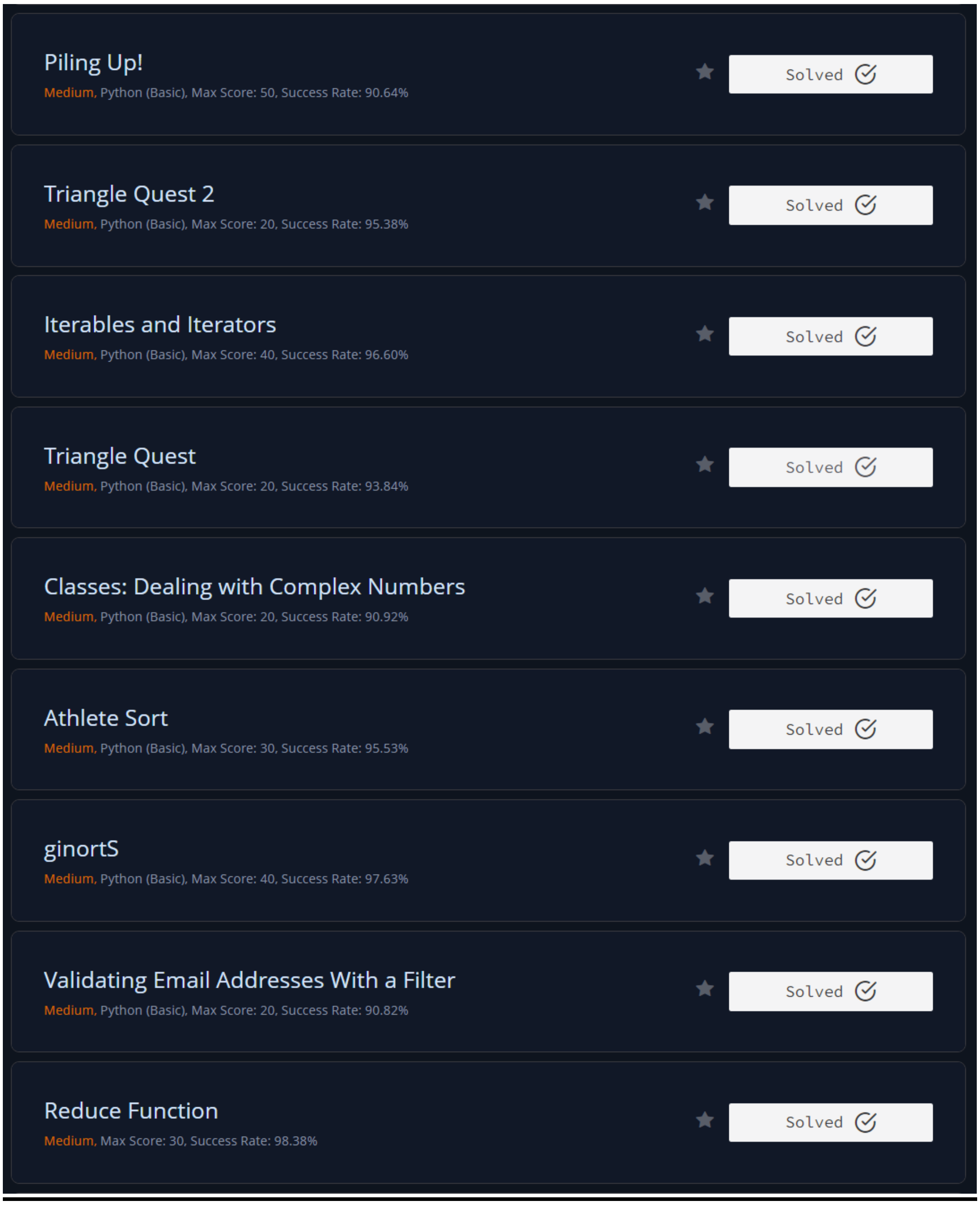
Find Angle MBC
Medium, Python (Basic), Max Score: 10, Success Rate: 89.16%

No Idea!
Medium, Python (Basic), Max Score: 50, Success Rate: 88.03%

Word Order
Medium, Python (Basic), Max Score: 50, Success Rate: 90.24%

Compress the String!
Medium, Python (Basic), Max Score: 20, Success Rate: 97.15%

Company Logo
Medium, Problem Solving (Basic), Max Score: 30, Success Rate: 89.84%



Piling Up!
Medium, Python (Basic), Max Score: 50, Success Rate: 90.64%
Solved

Triangle Quest 2
Medium, Python (Basic), Max Score: 20, Success Rate: 95.38%
Solved

Iterables and Iterators
Medium, Python (Basic), Max Score: 40, Success Rate: 96.60%
Solved

Triangle Quest
Medium, Python (Basic), Max Score: 20, Success Rate: 93.84%
Solved

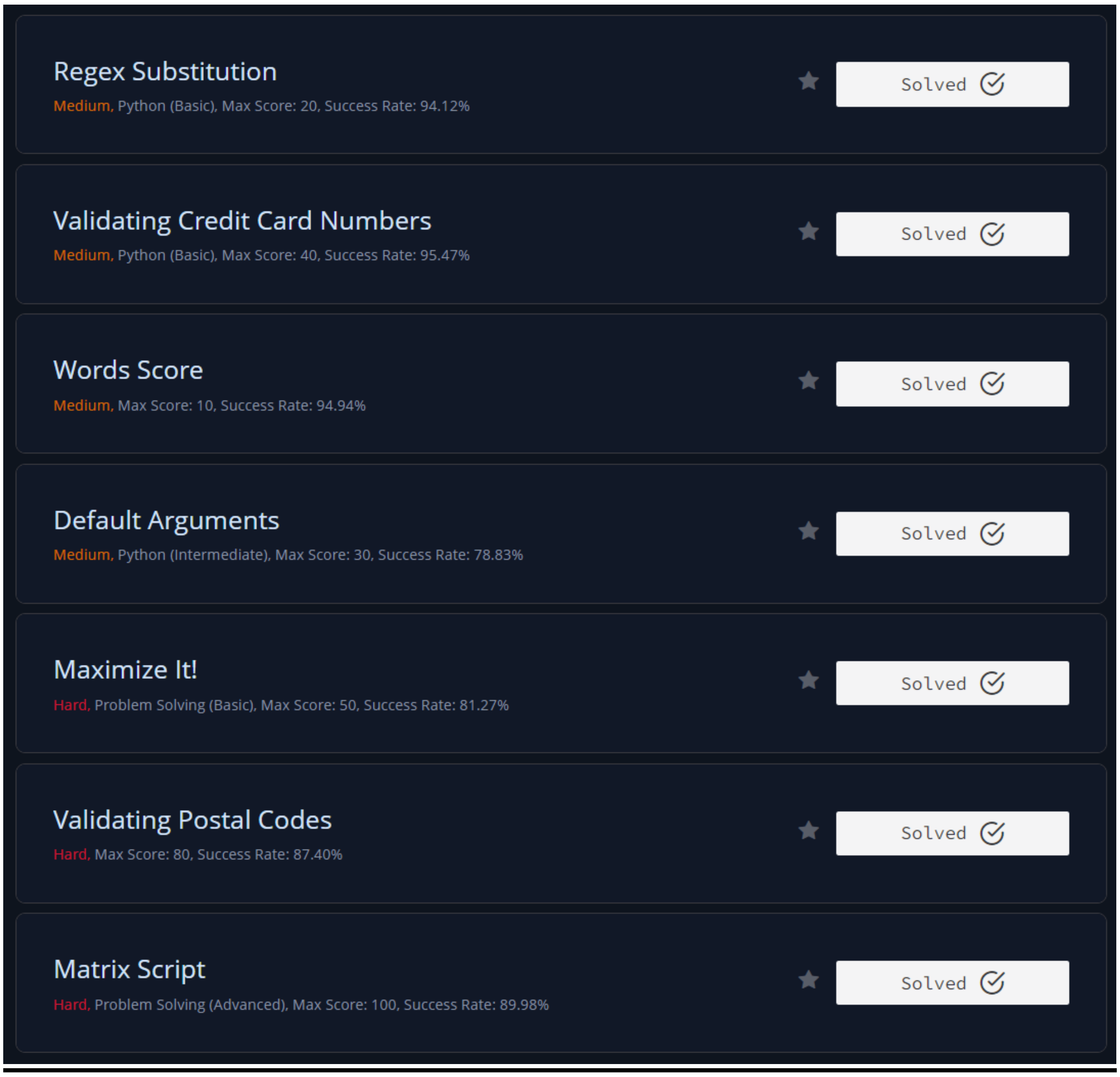
Classes: Dealing with Complex Numbers
Medium, Python (Basic), Max Score: 20, Success Rate: 90.92%
Solved

Athlete Sort
Medium, Python (Basic), Max Score: 30, Success Rate: 95.53%
Solved

ginortS
Medium, Python (Basic), Max Score: 40, Success Rate: 97.63%
Solved

Validating Email Addresses With a Filter
Medium, Python (Basic), Max Score: 20, Success Rate: 90.82%
Solved

Reduce Function
Medium, Max Score: 30, Success Rate: 98.38%
Solved



Regex Substitution

Medium, Python (Basic), Max Score: 20, Success Rate: 94.12%

Validating Credit Card Numbers

Medium, Python (Basic), Max Score: 40, Success Rate: 95.47%

Words Score

Medium, Max Score: 10, Success Rate: 94.94%

Default Arguments

Medium, Python (Intermediate), Max Score: 30, Success Rate: 78.83%

Maximize It!

Hard, Problem Solving (Basic), Max Score: 50, Success Rate: 81.27%

Validating Postal Codes

Hard, Max Score: 80, Success Rate: 87.40%

Matrix Script

Hard, Problem Solving (Advanced), Max Score: 100, Success Rate: 89.98%