

Name: Muhammad Yousuf Eisa
Roll no: 452506
Assignment no:3
Artificial Intelligence

Medium Level Challenges

1- Write a function

The screenshot shows the HackerRank interface for the 'Write a function' challenge. The problem description explains that a leap year is added to the calendar every four years as February 29, and it lists the conditions for a leap year: divisible by 4, not divisible by 100, or divisible by 400. The task is to write a function that returns True for a leap year and False otherwise. The test cases section shows five cases, all of which are passed. A green banner at the top right says 'Congratulations' and 'You solved this challenge. Would you like to challenge your friends?'. Below this, there is a 'Compiler Message' box that says 'Success'. At the bottom, there is a 'Hidden Test Case' section that says 'Unlock this testcase for 5 hacks.' and a button to 'Unlock'.

Code

```
def is_leap(year):  
    return (year % 400 == 0) or ((year % 4 == 0) and (year % 100 != 0))  
year = int(input())  
print(is_leap(year))
```

2- Minion Game

The screenshot shows the HackerRank interface for the 'The Minion Game' challenge. The problem description is visible on the left, and the right side shows the code editor and test results.

Problem Description:

For better understanding, see the image below:

STUART 🏆

WORDS	SCORE
B	1
N	2
BA	1
NA	2
BAN	1
NAN	1
BANA	1
NANA	1
BANAN	1
BANANA	1
TOTAL	10

Congratulations

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

[Next Challenge](#)

Test Results:

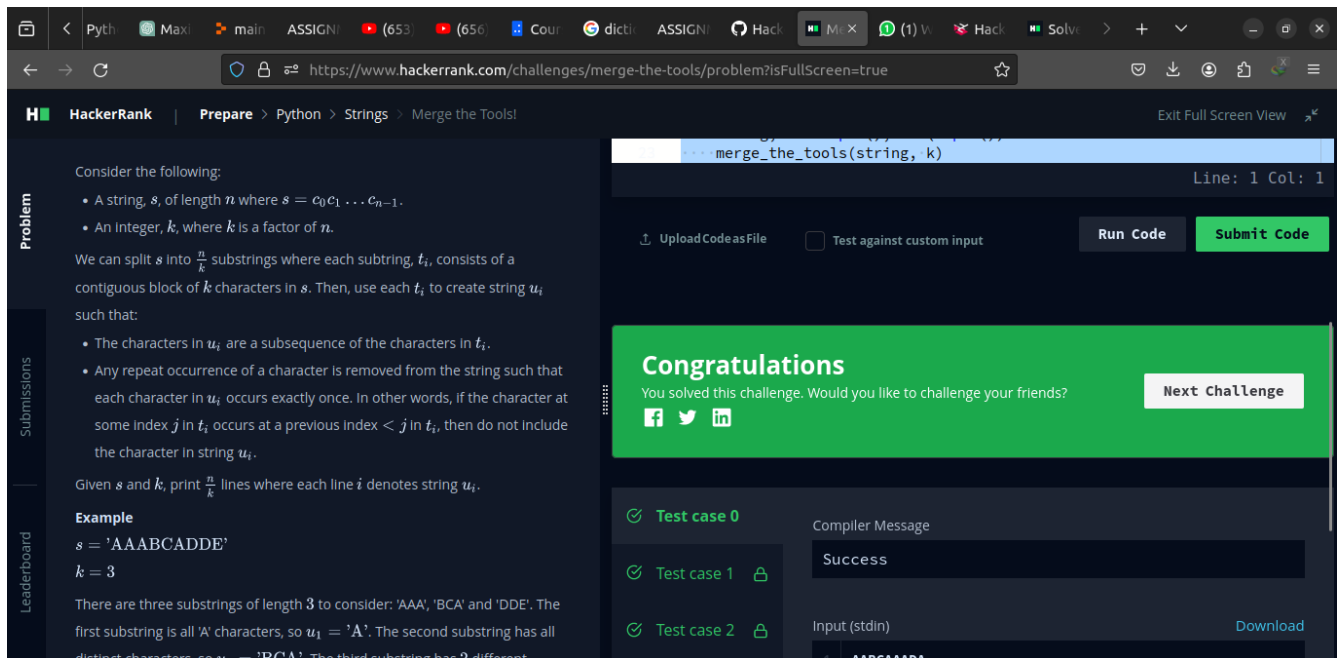
- Test case 0: Success
- Test case 1: Success
- Test case 2: Success

Code

vowels = ['A', 'E', 'I', 'O', 'U']

```
def minion_game(string):
    score_kevin = 0
    score_stuart = 0
    for ind in range(len(string)):
        if string[ind] in vowels:
            score_kevin += len(string) - ind
        else:
            score_stuart += len(string) - ind
    if score_kevin > score_stuart:
        print("Kevin {}".format(score_kevin))
    elif score_kevin < score_stuart:
        print("Stuart {}".format(score_stuart))
    else:
        print("Draw")
if __name__ == '__main__':
    s = input()
    minion_game(s)
```

3- Merge the tools



Code

```
def merge_the_tools(string, k):
    block_cnt = len(string)//k
    output_t = []
    output_u = []
    #print("{} / {} = {}".format(len(string), k, block_cnt))
    for ind in range(0, len(string) - k + 1, k):
        output_t.append(string[ind:ind + k])
        for block in output_t:
            for char in block:
                char_count = block.count(char)
                if char_count > 1:
                    block = block[::-1]
                    block = block.replace(char, "", char_count - 1)
                    block = block[::-1]
            output_u.append(block)
    print("\n".join(map(str, output_u)))

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

4- Time Delta

Problem

When users post an update on social media, such as a URL, image, status update etc., other users in their network are able to view this new post on their news feed. Users can also see exactly when the post was published, i.e. how many hours, minutes or seconds ago.

Since sometimes posts are published and viewed in different time zones, this can be confusing. You are given two timestamps of one such post that a user can see on his newsfeed in the following format:

Day dd Mon yyyy hh:mm:ss +xxxx

Here +xxxx represents the time zone. Your task is to print the absolute difference (in seconds) between them.

Input Format

The first line contains T , the number of testcases.

Each testcase contains 2 lines, representing time t_1 and time t_2 .

Constraints

- Input contains only valid timestamps
- $year \leq 3000$.

```
15 .....t2 = input().strip()
16 .....delta = time_delta(t1, t2)
17 .....print(delta)
```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input **Run Code** **Submit Code**

Congratulations

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

Next Challenge

Test case 0 **Test case 1** **Compiler Message**

Success

Code

```
import sys
```

```
from datetime import datetime as dt
```

```
dformat = "%a %d %b %Y %H:%M:%S %z"
```

```
def time_delta(t1, t2):
```

```
    first = dt.strptime(t1, dformat)
```

```
    second = dt.strptime(t2, dformat)
```

```
    return int(abs((first - second).total_seconds()))
```

```
if __name__ == "__main__":
```

```
    t = int(input().strip())
```

```
    for a0 in range(t):
```

```
        t1 = input().strip()
```

```
        t2 = input().strip()
```

```
        delta = time_delta(t1, t2)
```

```
        print(delta)
```

5- Angle MBC

HackerRank | Prepare > Python > Math > Find Angle MBC Exit Full Screen View

Line: 10 Col: 36

☐ Test against custom input

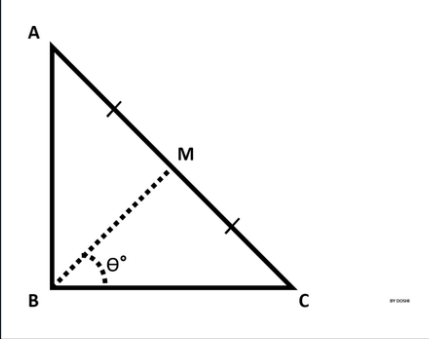
Congratulations
You solved this challenge. Would you like to challenge your friends?
[f](#) [t](#) [in](#)

✓ Test case 0 ✓ Test case 1 ✓ Test case 2

Compiler Message
Success

Input (stdin)
10 [Download](#)

Problem



ABC is a right triangle, 90° at *B*.
Therefore, $\angle ABC = 90^\circ$.
Point *M* is the midpoint of hypotenuse *AC*.
You are given the lengths *AB* and *BC*.
Your task is to find $\angle MBC$ (angle θ° , as shown in the figure) in degrees.

Code

```
from math import atan
from math import degrees

if __name__ == "__main__":
    ab = int(input().strip())
    bc = int(input().strip())
    print("{}°".format(int(round(degrees(atan(ab/bc))))))
```

6- No Ideal

The screenshot shows the HackerRank interface for the 'No Ideal' challenge. The problem description states: 'There is an array of n integers. There are also 2 disjoint sets, A and B , each containing m integers. You like all the integers in set A and dislike all the integers in set B . Your initial happiness is 0. For each i integer in the array, if $i \in A$, you add 1 to your happiness. If $i \in B$, you add -1 to your happiness. Otherwise, your happiness does not change. Output your final happiness at the end.' The constraints are: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$, and $1 \leq \text{Any integer in the input} \leq 10^9$. The input format specifies three lines: n and m , the array elements, and the sets A and B . The output format is a single integer representing the final happiness. The submission section shows 'Test case 0' as 'Success'.

Code

```
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(' ')))
    for el in arr:
        if el in good:
            happiness += 1
        elif el in bad:
            happiness -= 1
    print(happiness)
```

7- Word Order

The screenshot shows the HackerRank interface for the 'Word Order' challenge. The problem description states: 'You are given n words. Some words may repeat. For each word, output its number of occurrences. The output order should correspond with the input order of appearance of the word. See the sample input/output for clarification. Note: Each input line ends with a "\n" character. Constraints: $1 \leq n \leq 10^5$. The sum of the lengths of all the words do not exceed 10^6 . All the words are composed of lowercase English letters only. Input Format: The first line contains the integer, n . The next n lines each contain a word. Output Format: Output 2 lines. On the first line, output the number of distinct words from the input. On the second line, output the number of occurrences for each distinct word. The interface shows a 'Congratulations' message: 'You solved this challenge. Would you like to challenge your friends?'. Below this, the test case results are shown: 'Test case 0' is 'Success', 'Test case 1' is 'Success', and 'Test case 2' is 'Success'. The 'Compiler Message' section shows 'Success'. The 'Input (stdin)' section shows the input: '4', 'apple', 'banana', 'apple', 'orange'.

Code

```
from collections import OrderedDict
```

```
if __name__ == "__main__":
    num = int(input().strip())
    history = OrderedDict()
    for _ in range(num):
        word = str(input().strip().split())
        if word not in history.keys():
            history[word] = 1
        else:
            history[word] += 1
    print(len(history.keys()))
    print(" ".join(map(str, history.values())))
```

8- Compress the string

The screenshot shows the HackerRank interface for the 'Compress the String' challenge. The left sidebar contains the problem description, input/output formats, constraints, and sample input. The right sidebar shows a 'Congratulations' message and test case results.

Problem Description: In this task, we would like for you to appreciate the usefulness of the `groupby()` function of `itertools`. To read more about this function, [Check this out](#).

You are given a string S . Suppose a character 'c' occurs consecutively X times in the string. Replace these consecutive occurrences of the character 'c' with (X, c) in the string.

For a better understanding of the problem, check the explanation.

Input Format
A single line of input consisting of the string S .

Output Format
A single line of output consisting of the modified string.

Constraints
All the characters of S denote integers between 0 and 9.
 $1 \leq |S| \leq 10^4$

Sample Input
1222311

Test Case Results:

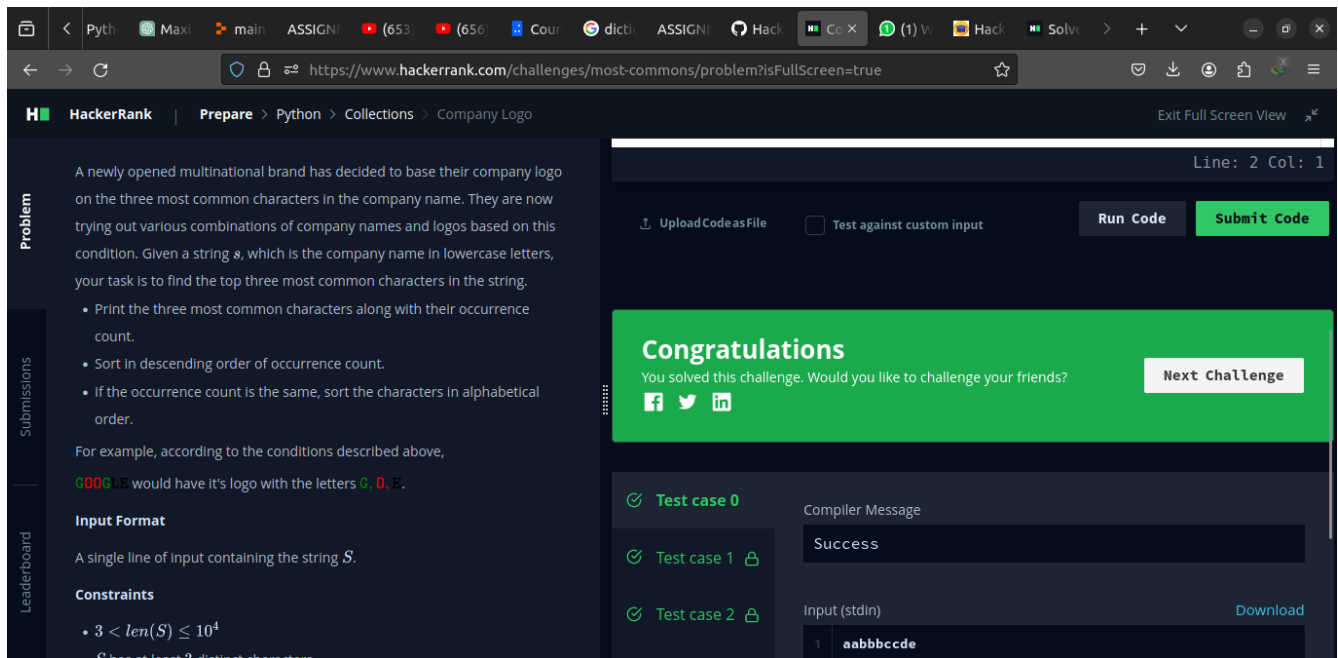
Test Case	Status	Compiler Message
Test case 0	Success	Success
Test case 1	Success	
Test case 2	Success	

Code

```
from itertools import groupby
```

```
if __name__ == "__main__":  
    #in_data = input().strip().split(' ')  
    for el, el_list in groupby(input()):  
        print((len(list(el_list)), int(el)), end=' ')
```


9- Company logo



The screenshot shows the HackerRank interface for the 'Company Logo' challenge. The problem description states: 'A newly opened multinational brand has decided to base their company logo on the three most common characters in the company name. They are now trying out various combinations of company names and logos based on this condition. Given a string s , which is the company name in lowercase letters, your task is to find the top three most common characters in the string.'

The input format is: 'A single line of input containing the string S .'

The constraints are: $3 < \text{len}(S) \leq 10^4$ and S has at least 3 distinct characters.

The submission section shows three test cases, all passed, with a 'Success' message and a 'Next Challenge' button.

Code

```
import math
import os
import random
import re
import sys

from collections import Counter, OrderedDict

class OrderedCounter(Counter, OrderedDict):
    pass
[print(*c) for c in OrderedCounter(sorted(input())).most_common(3)]
```

10- Pilling up

The screenshot shows the HackerRank interface for the 'Piling Up' challenge. The problem description states: 'There is a horizontal row of n cubes. The length of each cube is given. You need to create a new vertical pile of cubes. The new pile should follow these directions: if $cube[i]$ is on top of $cube[j]$ then $sideLength[j] \geq sideLength[i]$. When stacking the cubes, you can only pick up either the leftmost or the rightmost cube each time. Print Yes if it is possible to stack the cubes. Otherwise, print No.' The example input is `blocks = [1, 2, 3, 8, 7]` and the result is 'No'. The 'Test case 0' section shows 'Success'.

Code

from collections import deque

```
if __name__ == "__main__":
    t = int(input().strip())
    for _ in range(t):
        num_cnt = int(input().strip())
        deq = deque(list(map(int, input().strip().split(' '))))
        prev = max(deq[0], deq[-1])
        while deq:
            if prev >= deq[0] and prev >= deq[-1]:
                if deq[0] >= deq[-1]:
                    prev = deq.popleft()
                else:
                    prev = deq.pop()
            else:
                break
        if len(deq) == 0:
            print('Yes')
        else:
            print('No')
```

11- Triangle Quest 2

The screenshot shows the HackerRank interface for the 'Triangle Quest 2' challenge. The problem description on the left states: 'You are given a positive integer N . Your task is to print a palindromic triangle of size N . For example, a palindromic triangle of size 5 is:'. A sample output is shown as a preformatted text block:

```
1
121
12321
1234321
123454321
```

. Below this, a note explains that the first line is pre-written and the user must complete the code using exactly one print statement. The input format is a single line with the integer N . On the right, the submission area shows a 'Success' message for 'Test case 0', 'Test case 1', and 'Test case 2'. The input (stdin) is shown as '5'. A green 'Congratulations' banner is displayed, indicating the user has solved the challenge.

Code

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

12- Iterables and Iterators

The screenshot shows the HackerRank interface for the 'Iterables and Iterators' challenge. The problem description on the left states: 'The itertools module standardizes a core set of fast, memory efficient tools that are useful by themselves or in combination. Together, they form an iterator algebra making it possible to construct specialized tools succinctly and efficiently in pure Python. To read more about the functions in this module, check out their documentation here. You are given a list of N lowercase English letters. For a given integer K , you can select any K indices (assume 1-based indexing) with a uniform probability from the list. Find the probability that at least one of the K indices selected will contain the letter: 'a'. Input Format: The input consists of three lines. The first line contains the integer N , denoting the length of the list. The next line consists of N space-separated lowercase English letters, denoting the elements of the list. The third and the last line of input contains the integer K , denoting the number of indices to be selected.' On the right, the submission area shows a 'Success' message for 'Test case 0', 'Test case 1', and 'Test case 2'. The input (stdin) is shown as '4'. A green 'Congratulations' banner is displayed, indicating the user has solved the challenge.

Code

```
import string
```

```
symbols = string.ascii_lowercase
```

```
from itertools import combinations
```

```
if __name__ == "__main__":
```

```
    n = int(input().strip())
```

```
    arr = list(map(str, input().strip().split(' ')))
```

```
    times = int(input().strip())
```

```
    cmbts = list(combinations(sorted(arr), times))
```

```
    print("{:.4f}".format(len(list(filter(lambda a: a[0] == 'a', cmbts)))/(len(cmbts))))
```

13- Triangle-Quest-1

The screenshot shows the HackerRank interface for the 'Triangle Quest' challenge. The problem description states: 'You are given a positive integer N . Print a numerical triangle of height $N - 1$ like the one below:'. A sample output is shown as a numerical triangle with 5 rows: 1, 22, 333, 4444, 55555. The challenge instructions specify using only arithmetic operations, a single for loop, and a print statement, with a limit of two lines of code. The 'Input Format' is a single line containing integer N , and the 'Constraints' are $1 \leq N \leq 9$. The right side of the interface shows a 'Congratulations' message, social media sharing options, and a 'Next Challenge' button. Below this, the test cases are listed: 'Test case 0' and 'Test case 1', both marked as successful. The compiler message area shows 'Success'.

Code

```
for i in range(1,int(input())):
```

```
    print( int((i*(pow(10, i) - 1)) / 9 ))
```

14- Athlete Sort

You are given a spreadsheet that contains a list of N athletes and their details (such as age, height, weight and so on). You are required to sort the data based on the K^{th} attribute and print the final resulting table. Follow the example given below for better understanding.

Rank	Age	Height (in cm)	Rank	Age	Height (in cm)
1	32	190	5	24	176
2	35	175	4	26	195
3	41	188	1	32	190
4	26	195	2	35	175
5	24	176	3	41	188

Note that K is indexed from 0 to $M - 1$, where M is the number of attributes.

Note: If two attributes are the same for different rows, for example, if two athletes are of the same age, print the row that appeared first in the input.

Input Format

The first line contains N and M separated by a space.

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

Test case 0 Success
Test case 1 Success

Input (stdin)
5 3

Code

```
import math
import os
import random
import re
import sys
if __name__ == "__main__":
    n, m = input().strip().split(' ')
    n, m = [int(n), int(m)]
    arr = []
    for arr_i in range(n):
        arr_t = [int(arr_temp) for arr_temp in input().strip().split(' ')]
        arr.append(arr_t)
    k = int(input().strip())
    for el in sorted(arr, key = lambda x: x[k]):
        print(" ".join(map(str, el)))
```

15- ginorts

The screenshot shows the HackerRank interface for the 'ginorts' challenge. The problem description states: 'You are given a string S . S contains alphanumeric characters only. Your task is to sort the string S in the following manner: All sorted lowercase letters are ahead of uppercase letters. All sorted uppercase letters are ahead of digits. All sorted odd digits are ahead of sorted even digits.' The input format is a single line containing the string S . The constraints are $0 < \text{len}(S) < 1000$. The output format is the sorted string S .

The right side of the page shows a 'Congratulations' message: 'You solved this challenge. Would you like to challenge your friends?' with a 'Next Challenge' button. Below this, there is a list of test cases (Test case 0 to Test case 5) and a 'Compiler Message' section showing 'Success'. The input (stdin) is 'Sorting1234' and the expected output is 'ginortS1324'.

Code

```
if __name__ == "__main__":
    string = input().strip()
    print(*sorted(string, key = lambda x: (-x.islower(), x.isdigit() - x.isupper(), x in '02468', x)), sep='')
```

16- Validating Email address with a filter

The screenshot shows the HackerRank interface for the challenge 'Validating Email Addresses With a Filter'. The problem description explains that a filter takes a function returning True or False and applies it to a sequence, returning a list of members where the function returned True. A Lambda function can be used with filters. The example task is to make a list of the squares of integers from 0 to 9 (both included).

The code editor shows the following solution:

```
>> l = list(range(10))
>> l = list(map(lambda x:x*x, l))

Now, you only require those elements that are greater than 10 but less than 80.

>> l = list(filter(lambda x: x > 10 and x < 80, l))
```

The 'Congratulations' message states: 'You solved this challenge. Would you like to challenge your friends?' with a 'Next Challenge' button.

The test cases section shows four cases, all marked as 'Success'.

The input (stdin) section shows the following input:

```
3
lara@hackerrank.com
```

Code

```
import re
def fun(email):
    # return True if s is a valid email, else return False
    # pattern = '[^@]+@[^@]+\.[^@]{1,3}'
    pattern = '^([a-zA-Z][w-]*@[a-zA-Z0-9]+\.[a-zA-Z]{1,3})$'
    return re.match(pattern, email)
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)
```

17- Reduce Function

The screenshot shows the HackerRank interface for the 'Reduce Function' challenge. The problem description states: 'Given a list of rational numbers, find their product.' The concept explains that the `reduce()` function applies a function of two arguments cumulatively on a list of objects in succession from left to right to reduce it to one value. An example is given: you have a list, say `[1, 2, 3]` and you have to find its sum. The code editor shows the following code:

```
>>> reduce(lambda x, y : x + y, [1,2,3])
6
```

Below the code editor, there is a note: 'You can also define an initial value. If it is specified, the function will assume initial value as the value given, and then reduce. It is equivalent to adding the initial value at the beginning of the list. For example:'

```
>>> reduce(lambda x, y : x + y, [1,2,3], -3)
3
>>> from fractions import gcd
>>> reduce(gcd, [2,4,8], 3)
1
```

The right side of the page shows a 'Congratulations' message: 'You solved this challenge. Would you like to challenge your friends?' with a 'Next Challenge' button. Below this, there is a 'Compiler Message' section showing 'Success'.

Code

```
from fractions import Fraction
from functools import reduce
def product(fracs):
    t = 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)
```


18- Regex Substitution

The `re.sub()` tool (sub stands for substitution) evaluates a pattern and, for each valid match, it calls a method (or lambda).
The method is called for all matches and can be used to modify strings in different ways.
The `re.sub()` method returns the modified string as an output.

Learn more about `re.sub()`.

Transformation of Strings

Code

```
import re

#Squaring numbers
def square(match):
    number = int(match.group(0))
    return str(number**2)

print re.sub(r"\d+", square, "1 2 3 4 5 6 7 8 9")
```

Problem

Submissions

Leaderboard

Output

Challenge Details: HackerRank > Prepare > Python > Regex and Parsing > Regex Substitution

Change Theme Language: Python 3

Enter your code here. Read input from STDIN. Print output to STDOUT

```
2 import re
3
4 def change(match):
5     ... symb = match.group(0)
6     ...
7     ... if symb == "&&":
8         ... return "and"
9     ... elif symb == "|":
10        ... return "or"
11
12 n = int(input().strip())
13
14 for _ in range(n):
15     ... print(re.sub(r'(<=.)(&&|\|)|(?=.)', change, input()))
```

Line: 2 Col: 1

Upload Code as File ☐ **Test against custom input** **Run Code** **Submit Code**

Code

```
import re
```

```
def change(match):
    symb = match.group(0)
    if symb == "&&":
        return "and"
    elif symb == "||":
        return "or"
    n = int(input().strip())
    for _ in range(n):
        print(re.sub(r'(?<= )(&&|\|\|)(?= )', change, input()))
```

19- Validating Credit Card Numbers

The screenshot shows the HackerRank interface for the 'Validating Credit Card Numbers' challenge. The problem description states that Fredrick received N credit cards from ABCD Bank and asks for help in validating them using regex. The characteristics of a valid credit card are listed: it must start with 4, 5, or 6; contain exactly 16 digits; consist only of digits (0-9); have digits in groups of 4 separated by hyphens; not use other separators like spaces or underscores; and not have 4 or more consecutive repeated digits. Examples of valid numbers are provided: 4253625879615786, 442442442442444, and 5122-2368-7954-3214. The right side of the interface shows a 'Congratulations' message, indicating the challenge was solved, and a 'Next Challenge' button. Below this, test cases are shown as 'Success', and the compiler message is 'Success'. The input (stdin) is shown as '6'.

Code

import re

```
if __name__ == "__main__":
    t = int(input().strip())
    for _ in range(t):
        num = "".join(input())
        if (re.match(r'^[456]', num) and
            (re.match(r'([\d]{4}-){3}[\d]{4}$', num) or
             re.match(r'[\d]{16}', num)) and
            not re.search(r'(\d)\1{3,}', num.replace("-", ""))):
            print("Valid")
        else:
            print("Invalid")
```

20- Word Score

HackerRank | Prepare > Python > Debugging > Words Score

Exit Full Screen View

Problem

In this challenge, the task is to debug the existing code to successfully execute all provided test files.

Consider that vowels in the alphabet are a, e, i, o, u and y.

Function `score_words` takes a list of lowercase words as an argument and returns a score as follows:

The score of a single word is 2 if the word contains an even number of vowels. Otherwise, the score of this word is 1. The score for the whole list of words is the sum of scores of all words in the list.

Debug the given function `score_words` such that it returns a correct score. Your function will be tested on several cases by the locked template code.

Input Format

The input is read by the provided locked code template. In the first line, there is a single integer `n` denoting the number of words. In the second line, there are `n` space-separated lowercase words.

Constraints

```
def is_vowel(letter):  
    num_vowels += 1  
    if num_vowels % 2 == 0:  
        score += 2  
    else:  
        score += 1  
    return score
```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input **Run Code** **Submit Code**

Congratulations
You solved this challenge. Would you like to challenge your friends?
f t in

Next Challenge

Test case 0 ☒ Compiler Message

https://www.facebook.com/sharer/sharer.php?k...6&display=popup&ref=plugin&src=share_button

Code

```
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))
```

21- Default Arguments

The screenshot shows the HackerRank interface for the 'Default Arguments' challenge. The problem description explains that Python supports default argument values for functions. A sample function 'increment_by' is provided, which takes 'n' and 'increment' (defaulting to 1) as arguments and returns 'n + increment'. The challenge requires debugging the 'print_from_stream' function using the default value of one. The solution code is shown in the code editor, and the 'Congratulations' message indicates that the challenge was solved successfully.

```
def increment_by(n, increment=1):
    return n + increment
```

```
>>> increment_by(5, 2)
7
>>> increment_by(4)
5
>>>
```

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
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()):
    temp = stream.current
    for _ in range(n):
```

Code

Enter your code here. Read input from STDIN. Print output to STDOUT

```
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()):
```

```
temp = stream.current
```

```
for _ in range(n):
```

```
print(stream.get_next())
stream.current = temp
```

```
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())
```

22- Classes: Dealing with Complex Numbers

The screenshot shows the HackerRank interface for the challenge 'Classes: Dealing with Complex Numbers'. The page is divided into several sections:

- Problem:** Contains the challenge description and sample input/output. The description states: "For complex numbers with non-zero real (A) and complex part (B), the output should be in the following format: $A + Bi$. Replace the plus symbol (+) with a minus symbol (-) when $B < 0$. For complex numbers with a zero complex part i.e. real numbers, the output should be: $A + 0.00i$. For complex numbers where the real part is zero and the complex part (B) is non-zero, the output should be: $0.00 + Bi$ ". The sample input is:
2 1
5 6
- Submissions:** A section for viewing and submitting solutions.
- Leaderboard:** A section for viewing the leaderboard.
- Test Cases:** A section showing the results of the test cases. All three test cases (Test case 0, Test case 1, and Test case 2) are marked as 'Success'.
- Compiler Message:** A section showing the compiler output, which is 'Success'.
- Input (stdin):** A section showing the input for the test cases, which is '2 1'.

At the top of the page, there is a notification: "You have earned 20.00 points! 40/115 challenges solved. 35%". Below this, there is a green banner with the text "Congratulations" and "You solved this challenge. Would you like to challenge your friends?". A "Next Challenge" button is also present.

Code

```
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+self.imaginary*no.real
return Complex(r,i)

def __truediv__(self, no):
d=no.real**2+no.imaginary**2
n=self*Complex(no.real,-1*no.imaginary)
return Complex(n.real/d,n.imaginary/d)

def mod(self):
d=self.real**2+self.imaginary**2
return Complex(math.sqrt(d),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__':
```

“Hard Challenges”

1- Maximize it

The screenshot shows the HackerRank interface for the 'Maximize It!' challenge. The problem description states: You are given a function $f(X) = X^2$. You are also given K lists. The i^{th} list consists of N_i elements. You have to pick one element from each list so that the value from the equation below is maximized: $S = (f(X_1) + f(X_2) + \dots + f(X_k)) \% M$. X_i denotes the element picked from the i^{th} list. Find the maximized value S_{max} obtained. $\%$ denotes the modulo operator. Note that you need to take exactly one element from each list, not necessarily the largest element. You add the squares of the chosen elements and perform the modulo operation. The maximum value that you can obtain, will be the answer to the problem.

Input Format

The first line contains 2 space separated integers K and M . The next K lines each contains an Integer N_i , denoting the number of elements in the i^{th} list, followed by N_i space separated integers denoting the elements of the i^{th} list.

The right side of the interface shows a 'Congratulations' message: 'You solved this challenge. Would you like to challenge your friends?' with a 'Next Challenge' button. Below this, it shows 'Test case 0', 'Test case 1', and 'Test case 2' all passed, with a 'Success' message in the compiler output.

Code

```
import itertools
```

```
k, m = map(int, input().split())
```

```
main_ar = []
```

```
for _ in range(k):
```

```
ar = list(map(int, input().split()))
```

```
main_ar.append(ar[1:])
```

```
all_combination = itertools.product(*main_ar)
```

```
result = 0
```

```
for single_combination in all_combination:
```

```
result = max(sum(x * x for x in single_combination) % m, result)
```

```
print(result)
```

2- Validating Postal Address

The screenshot shows the HackerRank interface for the 'Validating Postal Codes' challenge. The problem description states that a valid postal code P must be a number in the range 100000 to 999999 inclusive and must not contain more than one alternating repetitive digit pair. Examples of alternating repetitive digits are provided. The task is to provide two regular expressions: `regex_integer_in_range` and `regex_alternating_repetitive_digit_pair`. The input field shows '110000'. The 'Submit Code' button is highlighted in green, and a 'Congratulations' message is displayed, indicating the challenge was solved.

Code

import re

```
num = input()
print(bool(re.match(r'^[1-9][\d]{5}$', num) and len(re.findall(r'(\d)(?=\d\1)', num))<2 ))
```


3- Matrix Script

The screenshot shows the HackerRank interface for the 'Matrix Script' challenge. The problem description states: 'Neo has a complex matrix script. The matrix script is a $N \times M$ grid of strings. It consists of alphanumeric characters, spaces and symbols (!,@,#,\$,%,&).' Below this, a 3x3 matrix is displayed with the following characters:

T	s	i
h	%	x
i		#
s	M	
\$	a	
#	t	%
i	r	!

Below the matrix, the decoded message is shown: 'This\$#is% Matrix# %!'. The right side of the interface shows a 'Congratulations' message and a 'Next Challenge' button.

Code

```
import re

n, m = input().strip().split(' ')
n, m = [int(n), int(m)]
matrix = []
for _ in range(n):
    matrix_t = str(input())
    matrix.append(matrix_t)
complete = ""
for el in zip(*matrix):
    complete += "".join(el)
print(re.sub(r'(?<=\w)([^\w]+)(?=\w)', " ", complete))
```

Submitted to Sir Muhammad Affan