

Q.1 Swap Case String

You are given a string and your task is to swap cases. In other words, convert all lowercase letters to uppercase letters and vice versa.

Sample Input 0

HackerRank.com presents "Pythonist 2".

Sample Output 0

hACKERrANK.COM PRESENTS "pYTHONIST 2"

```
In [3]: def swap_case(s):
        new_str=''
        for i in s:
            if (i.isupper())==True:
                new_str+=i.lower()
            elif(i.islower())==True:
                new_str+=i.upper()
            else:
                new_str+=i
        return new_str

        if __name__ == '__main__':
            print('Input String:')
            s = input()
            result = swap_case(s)
            print('\nOutput String:')
            print(result)
```

Input String:
HackerRank.com presents "Pythonist 2".

Output String:
hACKERrANK.COM PRESENTS "pYTHONIST 2".

Q.2 String Split and Join

You are given a string. Split the string on a " " (space) delimiter and join using a - hyphen.

Input Format

The first line contains a string consisting of space separated words.

Output Format

Print the formatted string as explained above.

Sample Input

this is a string

Sample Output

this-is-a-string

```
In [4]: def split_and_join(line):
        line=line.split(' ')
        line='-'.join(line)
        return line

        if __name__ == '__main__':
            print('Input String')
            line = input()
            result = split_and_join(line)
            print('\nOutput String')
            print(result)
```

Input String
this is a string

Output String
this-is-a-string

Q.3 What's Your Name?

You are given the firstname and lastname of a person on two different lines. Your task is to read them and

print the following:

Hello firstname lastname! You just delved into python.

Input Format

The first line contains the first name, and the second line contains the last name.

Output Format

Print the output as mentioned above.

Sample Input 0

```
Ross  
Taylor
```

Sample Output 0

Hello Ross Taylor! You just delved into python.

```
In [5]: def print_full_name(first, last):  
        print(f'Hello {first} {last}! You just delved into python.')  
        if __name__ == '__main__':  
            first_name = input()  
            last_name = input()  
            print_full_name(first_name, last_name)
```

```
Ross  
Taylor  
Hello Ross Taylor! You just delved into python.
```

Q.4 String Mutations

Read a given string, change the character at a given index and then print the modified string.

Input Format

The first line contains a string, S. The next line contains an integer i , denoting the index location and a character c separated by a space.

Output Format

Using any of the methods explained above, replace the character at index i with character c .

Sample Input

abracadabra

5 k

Sample Output

abrackdabr

```
In [1]: def mutate_string(string, position, character):  
        return string[:position]+str(character)+string[position+1:]  
  
        if __name__ == '__main__':  
            print('Input String')  
            s = input()  
            print('\nIndex and Character')  
            i, c = input().split()  
            s_new = mutate_string(s, int(i), c)  
            print('\nOutput String')  
            print(s_new)
```

Input String
abracadabra

Index and Character
5 k

Output String
abrackdabra

Find a String

In this challenge, the user enters a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

NOTE: String letters are case-sensitive.

Input Format

The first line of input contains the original string. The next line contains the substring.

Each character in the string is an ascii character.

Output Format

Output the integer number indicating the total number of occurrences of the substring in the original string.

Sample Input

```
ABCD CDC
CDC
```

Sample Output

```
2
```

```
In [1]: def count_substring(string, sub_string):
        count=0
        for i in range (len(string)):
            if string[i:i+len(sub_string)]==sub_string:
                count += 1
        return count

        if __name__ == '__main__':
            print('Enter String:')
            string = input().strip()
```

```
print('\nEnter Sub-string:')
sub_string = input().strip()

count = count_substring(string, sub_string)
print('\nCount of Sub string in string:')
print(count)
```

Enter String:
ABCD CDC

Enter Sub-string:
CDC

Count of Sub string in string:
2

Q.6 String Validator

You are given a string S . Your task is to find out if the string S contains: alphanumeric characters, alphabetical characters, digits, lowercase and uppercase characters.

Input Format

A single line containing a string S .

Output Format

In the first line, print True if S has any alphanumeric characters. Otherwise, print False .
In the second line, print True if S has any alphabetical characters. Otherwise, print False .
In the third line, print True if S has any digits. Otherwise, print False .
In the fourth line, print True if S has any lowercase characters. Otherwise, print False .
In the fifth line, print True if S has any uppercase characters. Otherwise, print False .

Sample Input

qA2

Sample Output

True
True
True
True
True

```
In [2]: print('Input String:')
s = input()
print('\nOutput Result\n')
print(any(i.isalnum() for i in s) )
print(any(i.isalpha() for i in s) )
print(any(i.isdigit() for i in s) )
print(any(i.islower() for i in s) )
print(any(i.isupper() for i in s) )
```

Input String:
qA2

Output Result

True
True
True
True
True

Q.7 Text Alignment

You are given a partial code that is used for generating the HackerRank Logo of variable thickness. Your task is to replace the blank (__) with rjust, ljust or center.

Input Format

A single line containing the thickness value for the logo.

Constraints

The thickness must be an odd number.

Output Format

Output the desired logo.

Sample Input

5

Sample Output

[illegible]

HHHHH
HHHHH
HHHHH

HHHHH
HHHHH
HHHHH
HHHHHHHHH
HHHHHHH
HHHHH
HHH
H

Q.8 Text Wrap

You are given a string S and width w . Your task is to wrap the string into a paragraph of width w .

Input Format

The first line contains a string, S . The second line contains the width, w .

Output Format

Print the text wrapped paragraph.

Sample Input 0

```
ABCDEFGHIJKLMNOQRSTUVWXYZ  
4
```

Sample Output 0

```
ABCD  
EFGH  
IJKL  
IMNO  
QRST  
UVWX  
Y
```

```
In [7]: def wrap(string, max_width):
```

```

str = ""
for i in range(0, len(string), max_width):
    str += string[i:i+max_width] + "\n"
return str

print('Enter String')
string = input()
print('\nEnter Max Width')
max_width = int(input())

result = wrap(string, max_width)
print('\nOutput\n')
print(result)

```

Enter String
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

Enter Max Width
 4

Output

ABCD
 EFGH
 IJKL
 IMNO
 QRST
 UVWX
 YZ

Q.9 Designer Door Mat

Mr. Vincent works in a door mat manufacturing company. One day, he designed a new door mat with the following specifications:

Mat size must be N X M (N is an odd natural number, N and 3 is N times .)

The design should have 'WELCOME' written in the center.

The design pattern should only use |, . and - characters.

Input Format

A single line containing the space separated values of and .

Output Format

Output the design pattern.

Sample Input

9 27

Sample Output

```
-----|.-----
-----|.|.|.-----
---|.|.|.|.|.|.-----
---|.|.|.|.|.|.|.-----
-----WELCOME-----
---|.|.|.|.|.|.|.-----
-----|.|.|.|.-----
-----|.|.|.-----
-----|.-----
```

```
In [1]: print('Input')
N,M=map(int,input().split())
pattern = [('.|.'*(2*i + 1)).center(M, '-') for i in range(N//2)]
print('\nOutput\n')
print('\n'.join(pattern + ['WELCOME'.center(M, '-')] + pattern[::-1]))
```

Input
7 21

Output

```
-----|.-----
-----|.|.|.-----
---|.|.|.|.|.|.-----
---|.|.|.|.|.|.|.-----
-----WELCOME-----
---|.|.|.|.|.|.|.-----
-----|.|.|.|.-----
-----|.-----
```

Q.10 String Formatting

Given an integer, n , print the following values for each integer from i to $1:n$

1. Decimal
2. Octal
3. Hexadecimal (capitalized)
4. Binary

The four values must be printed on a single line in the order specified above for each i from 1 to n . Each value should be space-padded to match the width of the binary value of n .

Input Format

A single integer denoting n .

Output Format

Print lines where each line i (in the range) contains the respective decimal, octal, capitalized hexadecimal, and binary values of i . Each printed value must be formatted to the width of the binary value of n .

Sample Input

17

Sample Output

```
1 1 1 1
2 2 2 10
3 3 3 11
4 4 4 100
5 5 5 101
6 6 6 110
7 7 7 111
```

```

8 10 8 1000
9 11 9 1001
10 12 A 1010
11 13 B 1011
12 14 C 1100
13 15 D 1101
14 16 E 1110
15 17 F 1111
16 20 10 10000
17 21 11 10001

```

```

In [3]: def print_formatted(number):
        width=len(str(bin(number))[2:])
        print('\nOutput\n')
        for i in range (1,number+1):
            print(str(i).rjust(width,' '),oct(i)[2:].rjust(width,' '),
                  hex(i)[2:].upper().rjust(width,' '),bin(i)[2:].rjust(width,' '))

        if __name__ == '__main__':
            print('Input\n')
            n = int(input())
            print_formatted(n)

```

Input

17

Output

```

1      1      1      1
2      2      2     10
3      3      3     11
4      4      4    100
5      5      5    101
6      6      6    110
7      7      7    111
8     10      8   1000
9     11      9   1001
10     12      A   1010
11     13      B   1011
12     14      C   1100
13     15      D   1101
14     16      E   1110
15     17      F   1111

```

16	20	10	10000
17	21	11	10001

Q.11 AlphaBet Rangoli

You are given an integer, N. Your task is to print an alphabet rangoli of size N .(Rangoli is a form of Indian folk art based on creation of patterns.)

Input Format

Only one line of input containing N, the size of the rangoli.

Output Format

Print the alphabet rangoli in the format explained above.

Sample Input

5

Sample Output

```

-----e-----
-----e-d-e-----
----e-d-c-d-e----
--e-d-c-b-c-d-e--
e-d-c-b-a-b-c-d-e
--e-d-c-b-c-d-e--
----e-d-c-d-e----
-----e-d-e-----
-----e-----

```

```

In [1]: def print_rangoli(size):
        n=size
        l=list(map(chr,range(97,123)))
        m=l[n-1::-1]+l[1:n]
        w=len('.'.join(m))
        for i in range(1,n):
            print('.'.join(l[n-1:n-i:-1]+l[n-i:n]).center(w,'.'))

```

```

for i in range (n,0,-1):
    print('-'.join(l[n-1:n-i:-1]+l[n-i:n]).center(w, '-'))

if __name__ == '__main__':
    n = int(input())
    print_rangoli(n)

```

```

5
-----e-----
-----e-d-e-----
----e-d-c-d-e----
--e-d-c-b-c-d-e--
e-d-c-b-a-b-c-d-e
--e-d-c-b-c-d-e--
----e-d-c-d-e----
-----e-d-e-----
-----e-----

```

Q.12 Capitalize

You are given a string, and your task is to capitalize each word of the string.

AFTER: You are asked to verify that the first, middle and last names of people begin with a capital letter in their passports. Given full names, your task is to capitalize the names appropriately.

Input Format

A single line of input containing the string,S.

Output Format

Print the capitalized string,S.

Sample Input

chris alan

Sample Output

Chris Alan

```
In [3]: def solve(s):
        l=s.split(' ')
        str=''
        for i in l:
            str=str+i.capitalize()+' '
        return str

        if __name__ == '__main__':

            s = input()

            result = solve(s)

            print(result)
```

```
chris alan
Chris Alan
```

Q13 The Minions Game

Game Rules

Both players are given the same string,S. Both players have to make substrings using the letters of the string S. Stuart has to make words starting with consonants. Kevin has to make words starting with vowels. The game ends when both players have made all possible substrings.

Scoring

A player gets +1 point for each occurrence of the substring in the string S .

For Example:

```
String S = BANANA
Kevin's vowel beginning word = ANA
```

Here, ANA occurs twice in BANANA. Hence, Kevin will get 2 Points.

For better understanding, see the image below

BANANA			
STUART 		KEVIN	
WORDS	SCORE	WORDS	SCORE
B	1	A	3
N	2	AN	2
BA	1	ANA	2
NA	2	ANAN	1
BAN	1	ANANA	1
NAN	1		
BANA	1		
NANA	1		
BANAN	1		
BANANA	1		
TOTAL	12	TOTAL	9

Your task is to determine the winner of the game and their score.

Input Format

A single line of input containing the string S.

Note: The string S will contain only uppercase letters[A:Z].

Output Format

Print one line: the name of the winner and their score separated by a space. If the game is a draw, print Draw.

Sample Input

BANANA

Sample Output

Stuart 12

```
In [4]: def minion_game(string):
        Stuart=0
        Kevin=0
        vowel='AEIOU'
        for i in range (len(string)):
            if string[i]not in vowel:
                Stuart=Stuart+(len(string)-i)
            else:
                Kevin=Kevin+(len(string)-i)

        print('\nWinner')
        if Stuart > Kevin:
            print('Stuart',Stuart)
        elif Kevin > Stuart:
            print('Kevin',Kevin)
        else:
            print('Draw')

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

Input String:
BANANA

Winner
Stuart 12

Q 14 Merge the Tools!

Sample Input

AABCAAADA
3

Sample Output

AB
CA
AD

```
In [8]: def merge_the_tools(string, k):
        l=[]
        m=0
        for i in range(len(string)//k):
            l.append(string[m:m+k])
            m+=k
        for j in l:
            print('\n')
            print(list(j))
            print(dict.fromkeys(list(j)))
            print(dict.fromkeys(list(j),1))
            print(dict.fromkeys(list(j)).keys())
            print(list(dict.fromkeys(list(j)).keys()))
            print('\n Output')
            print(''.join(list(dict.fromkeys(list(j)).keys()))

        # your code goes here

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

Input
AABCAAADA
3

```
['A', 'A', 'B']
{'A': None, 'B': None}
{'A': 1, 'B': 1}
dict_keys(['A', 'B'])
```

```
['A', 'B']
```

```
Output  
AB
```

```
['C', 'A', 'A']  
{'C': None, 'A': None}  
{'C': 1, 'A': 1}  
dict_keys(['C', 'A'])  
['C', 'A']
```

```
Output  
CA
```

```
['A', 'D', 'A']  
{'A': None, 'D': None}  
{'A': 1, 'D': 1}  
dict_keys(['A', 'D'])  
['A', 'D']
```

```
Output  
AD
```

In []: