Programming Task - 4

```
# sWAP cASE in Python - HackerRank Solution
def swap_case(s):

    # sWAP cASE in Python - HackerRank Solution START
    Output = '';
    for char in s:
        if(char.isupper()==True):
            Output += (char.lower());
        elif(char.islower()==True):
            Output += (char.upper());
        else:
            Output += char;
    return Output;
    # sWAP cASE in Python - HackerRank Solution END




if __name__ == '__main__':
    s = input()
    result = swap_case(s)
    print(result)
```

```
 ramffg
 RAMFFG
```

```

# Complete the 'print_full_name' function below.
#
# The function is expected to return a STRING.
# The function accepts following parameters:
#  1. STRING first
#  2. STRING last
#

def print_full_name(a, b):

    # What's Your Name? in Python - HackerRank Solution START
    b = b+"!"
    print("Hello",a, b,"You just delved into python.");

if __name__ == '__main__':
    first_name = input()
    last_name = input()
    print_full_name(first_name, last_name)
```

```
 ram
 hf
 Hello ram hf! You just delved into python.
```

```


def minion_game(string):
    # your code goes here
    # The Minion Game in Python - Hacker Rank Solution START
    player1 = 0;
    player2 = 0;
    str_len = len(string)
    for i in range(str_len):
        if s[i] in "AEIOU":
            player1 += (str_len)-i
        else :
            player2 += (str_len)-i

    if player1 > player2:
        print("Kevin", player1)
    elif player1 < player2:
        print("Stuart",player2)
    elif player1 == player2:
        print("Draw")
    else :
        print("Draw")
    # The Minion Game in Python - Hacker Rank Solution END
```

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

```
import textwrap

def wrap(string, max_width):
    # Text Wrap in Python - HackerRank Solution START
    return textwrap.fill(string,max_width)
    # Text Wrap in Python - HackerRank Solution END

if __name__ == '__main__':
    string, max_width = input(), int(input())
    result = wrap(string, max_width)
    print(result)
```

```
#Replace all ______ with rjust, ljust or center.

thickness = int(input()) #This must be an odd number
c = 'H'

#Top Cone
# replace  ______ To rjust |  ______ To  ljust
for i in range(thickness):
    print((c*i).rjust(thickness-1)+c+(c*i).ljust(thickness-1))

#Top Pillars
# replace ______ To center |  ______ To center
for i in range(thickness+1):
    print((c*thickness).center(thickness*2)+(c*thickness).center(thickness*6))

#Middle Belt
# replace ______ To center
for i in range((thickness+1)//2):
    print((c*thickness*5).center(thickness*6))

#Bottom Pillars
# replace ______ To center |  ______ To center
for i in range(thickness+1):
    print((c*thickness).center(thickness*2)+(c*thickness).center(thickness*6))

#Bottom Cone
# replace ______ To rjust |  ______ To ljust |  ______ To rjust
for i in range(thickness):
    print(((c*(thickness-i-1)).rjust(thickness)+c+(c*(thickness-i-1)).ljust(thickness)).rjust(thickness*6))

''' Text Allignment in Python - HackerRank Solution END '''
```

```
if __name__ == '__main__':
    S = input()
print(any(char.isalnum() for char in S))
print(any(char.isalpha() for char in S))
print(any(char.isdigit() for char in S))
print(any(char.islower() for char in S))
print(any(char.isupper() for char in S))
```

```
def split_and_join(line):
    a = line.split(" ")
    a = "-".join(a)
    return a




if __name__ == '__main__':
    line = input()
    result = split_and_join(line)
    print(result)
```

```
# String Formatting in Python - HackerRank Solution
def print_formatted(number):
    # your code goes here
    # String Formatting in Python - HackerRank Solution START
    l1 = len(bin(number)[2:])
    for i in range(1, number + 1):
        print(str(i).rjust(l1, ' '), end=" ")
```

```
        print(oct(i)[2:].rjust(l1, ' '), end=" ")
        print(((hex(i)[2:]).upper()).rjust(l1, ' '), end=" ")
        print(bin(i)[2:].rjust(l1, ' '), end=" ")
        print("")



if __name__ == '__main__':
    n = int(input())
    print_formatted(n)
```

```
def mutate_string(string, position, character):
    # Mutations in Python - HackerRank Solution START
    l = list(string)
    l[position] = character;
    string = ''.join(l);
    return string
    # Mutations in Python - HackerRank Solution END


if __name__ == '__main__':
    s = input()
    i, c = input().split()
    s_new = mutate_string(s, int(i), c)
    print(s_new)
Footer
© 2023 GitHub, Inc.
Footer navigation
Terms
Privacy
```

```
def merge_the_tools(string, k):
    # your code goes here
    for i in range(0,len(string), k):
        #slice string upto k characters
        line = string[i:i+k]
        seen = set()
        for i in line:
            #only print if we haven't already seen this character
            if i not in seen:
                print(i,end="")
                seen.add(i)
        #prints a new line
        print()


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

```
# Find a string in Python - HackerRank Solution
def count_substring(string, sub_string):
    # Find a string in Python - Hacker Rank Solution START
    count = 0
    for i in range(len(string)-len(sub_string)+1):
        if (string[i:i+len(sub_string)] == sub_string):
            count += 1
    return count
    # Find a string in Python - HackerRank Solution END



if __name__ == '__main__':
    string = input().strip()
    sub_string = input().strip()

    count = count_substring(string, sub_string)
    print(count)
```

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
# Designer Door Mat in Python - HackerRank Solution START
N, M = map(int, input().split())
for i in range(1, N, 2):
    print(str('.|.' * i).center(M, '-'))
print('WELCOME'.center(M, '-'))
for i in range(N-2, -1, -2):
    print(str('.|.' * i).center(M, '-'))
# Designer Door Mat in Python - HackerRank Solution END
```

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



# Complete the solve function below.
def solve(s):
    for i in s.split():
        s = s.replace(i,i.capitalize())
    return s


if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    s = input()

    result = solve(s)

    fptr.write(result + '\n')

    fptr.close()
```

```
def print_rangoli(size):
    # your code goes here
    width  = size*4-3
    string = ''

    for i in range(1,size+1):
        for j in range(0,i):
            string += chr(96+size-j)
            if len(string) < width :
                string += '-'
        for k in range(i-1,0,-1):
            string += chr(97+size-k)
            if len(string) < width :
                string += '-'
        print(string.center(width,'-'))
        string = ''

    for i in range(size-1,0,-1):
        string = ''
        for j in range(0,i):
            string += chr(96+size-j)
            if len(string) < width :
                string += '-'
        for k in range(i-1,0,-1):
            string += chr(97+size-k)
            if len(string) < width :
                string += '-'
        print(string.center(width,'-'))

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

✓ 35s completed at 4:22 PM