**Strings**

**CamelCase**

import math

import os

import random

import re

import sys

def camelcase(s):

    ctr = 1

    for char in s:

        if char.isupper():

            ctr+=1

    return ctr

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    s = input()

    result = camelcase(s)

    fptr.write(str(result) + '\n')

    fptr.close()

# Strong Password

import math

import os

import random

import re

import sys

def minimumNumber(n, password):

    count = 0

    if any(i.isdigit() for i in password)==False:

        count+=1

    if any(i.islower() for i in password)==False:

        count+=1

    if any(i.isupper() for i in password)==False:

        count+=1

    if any(i in '!@#$%^&\*()-+' for i in password)==False:

        count+=1

    return max(count,6-n)

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    n = int(input())

    password = input()

    answer = minimumNumber(n, password)

    fptr.write(str(answer) + '\n')

    fptr.close()

# Mars Exploration

import math

import os

import random

import re

import sys

def marsExploration(s):

    count= 0

    r=len(s)//3

    target= "SOS"\*r

    for i,j in zip(s, target):

        if i!=j:

            count+=1

    return count

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    s = input()

    result = marsExploration(s)

    fptr.write(str(result) + '\n')

    fptr.close()

# HackerRank in a String!

import math

import os

import random

import re

import sys

def hackerrankInString(s):

    target = "hackerrank"

    n = len(target)

    i=0

    for char in s:

        if char == target[i]:

            i+=1

            if i==n:

                return "YES"

    return "NO"

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    q = int(input())

    for q\_itr in range(q):

        s = input()

        result = hackerrankInString(s)

        fptr.write(result + '\n')

    fptr.close()

# Pangrams

import math

import os

import random

import re

import sys

def pangrams(s):

    temp= set(s.lower())-set(' ')

    if len(temp)==26:

        return "pangram"

    else:

        return "not pangram"

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    s = input()

    result = pangrams(s)

    fptr.write(result + '\n')

    fptr.close()