In [2]:

```
#1) swap case
def swap_case(s):

    x=""
    for c in s:
        if c.isupper():
            c=c.lower()
        else:
            c=c.upper()
        x+="".join(c)
    return x



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

```
HackerRank.com presents "Pythonist 2".
hACKERrANK.COM PRESENTS "pYTHONIST 2".
```

In [3]:

```
#2) String Split and Join

def split_and_join(line):
    # write your code here
    a=line.split()
    x="-".join(a)
    return x


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

```
this is a string
this-is-a-string
```

In [4]:

```
#3) What's Your Name?
def print_full_name(first, last):
    # Write your code here
    print("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.
```

In [5]:

```
#4) Mutations

def mutate_string(string, position, character):


    x = list(string)
    x[position] = character
    string = ''.join(x)
```

```
    return string



if __name__ == '__main__':

    s = input()
    i, c = input().split()
    s_new = mutate_string(s, int(i), c)
    print(s_new)
```

```
abracadabra
5 k
abrackdabra
```

In [6]:

```
#5) Find a string

def count_substring(string, sub_string):
    c=0
    x=len(string)
    y=len(sub_string)


    for i in range(x-y+1):
        if(string[i:(i+y)] == sub_string):
            c=c+1
    return c

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

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

```
ABCDCDC
CDC
2
```

In [7]:

```
#6) String Validators

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))
```

```
qA2
True
True
True
True
True
```

In [8]:

```
#7) Text Alignment

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

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

#Top Pillars
```

```
for i in range(thickness+1):
    print((c*thickness).center(thickness*2)+(c*thickness).center(thickness*6))

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

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

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

```
5
    H
   HHH
  HHHHH
 HHHHHHH
HHHHHHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHHHHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHHHHHHH
  HHHHHHHHHHHHHHHHHHHHHHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
  HHHHH                 HHHHH
                    HHHHHHHHH
                     HHHHHHH
                      HHHHH
                       HHH
                        H
```

In [9]:

```
#8) Text Wrap

import textwrap

def wrap(string, max_width):
    return textwrap.fill(string,max_width)

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

```
ABCDEFGHIJKLIMNOQRSTUVWXYZ
4
ABCD
EFGH
IJKL
IMNO
QRST
UVWX
YZ
```

In [10]:

```
#9) Designer Door Mat

N,M=map(int,input().split(' '))
```

```
for i in range(1,N,2):
    print((".|."*i).center(M,'-'))

print("WELCOME".center(M,'-'))

for i in range(N-2,-1,-2):
    print((".|."*i).center(M,'-'))
```

```
9 27
------------.|.------------
---------.|..|..|.---------
------.|..|..|..|..|.------
---.|..|..|..|..|..|..|.---
----------WELCOME----------
---.|..|..|..|..|..|..|.---
------.|..|..|..|..|.------
---------.|..|..|.---------
------------.|.------------
```

In [11]:

```
#10) String Formatting

def print_formatted(number):

    x = len(bin(number)[2:])
    for i in range(1, number+1):
        deci = str(i)
        octa = oct(i)[2:]
        hexa = hex(i)[2:].upper()
        bina = bin(i)[2:]

        print(deci.rjust(x),octa.rjust(x),hexa.rjust(x),bina.rjust(x))


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

```
17
    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 [12]:

```
#11) Alphabet Rangoli

def print_rangoli(size):
    # your code goes here
    import string
    alpha = string.ascii_lowercase

    x= []
    for i in range(a):
        s = "-".join(alpha[i:a])
        L.append((s[::-1]+s[1:]).center(a*4-3, "-"))
```

```
    print('\n'.join(x[:0:-1]+x))

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--------
```

In [14]:

```
#12) Capitalize!
#working well in hackerrank
#input- chris alan
#ouput- Chris Alan


import math
import os
import random
import re
import sys

# Complete the solve function below.
def solve(s):
    x=s.split(" ")
    s=''
    for i in x:
        s=s+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()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-14-2485bef03a67> in <module>
     16
     17 if __name__ == '__main__':
---> 18     fptr = open(os.environ['OUTPUT_PATH'], 'w')
     19
     20     s = input()

~\anaconda3\lib\os.py in __getitem__(self, key)
    673         except KeyError:
    674             # raise KeyError with the original key value
--> 675             raise KeyError(key) from None
    676         return self.decodevalue(value)
    677

KeyError: 'OUTPUT_PATH'
```

In [15]:

```
#13) The Minion Game
```

```
def minion_game(string):
    # your code goes here
    s=len(string)
    a,b=0,0
   # a is constant and b is vowel
    for i in range(s):
        if string[i] in 'AEIOU':
            b=b+(s-i)
        else:
            a=a+(s-i)
    if a>b:
        print('Stuart {}'.format(a))
    elif a==b:
        print('Draw')
    else:
        print('Kevin {}'.format(b))
if __name__ == '__main__':
    s = input()
    minion_game(s)
```

```
BANANA
Stuart 12
```

In [16]:

```
#14) Merge the Tools!

def merge_the_tools(string, k):
    # your code goes here
    a=0
    s=''
    for i in string:
        if i not in s:
            s=s+i
        a+=1
        if(a==k):
            print(s)
            a=0
            s=""
if __name__ == '__main__':
    string, k = input(), int(input())
    merge_the_tools(string, k)
```

```
AABCAAADA
3
AB
CA
AD
```

In [ ]: