Name : SK Shivaanee

Roll no : 2310257

Section : S02 **Assignment 7 : Programs using Strings without using in-built functions**

Aim: To learn python programming using Strings without using in-built functions by forming expressions and statements involving reading and printing the data appropriately for the given specification.

Solve the following problems using Python (CO3, K3):

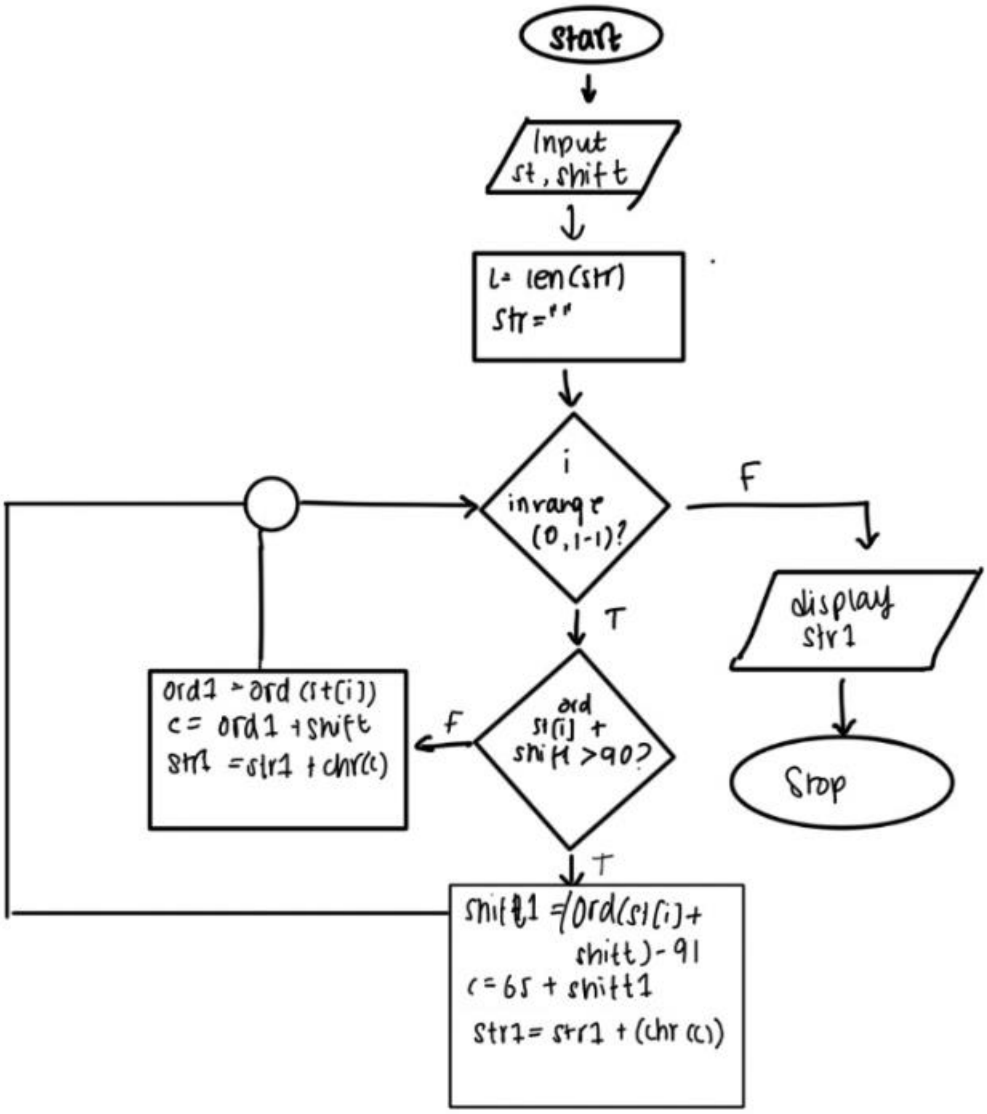
Question 1: In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions up the alphabet. The method is named after Julius Caesar, who used it in his private correspondence. In the movie 2001: A Space Odyssey, the ship computer is called HAL, which is IBM rotated by -1.

Consider the following sample input:

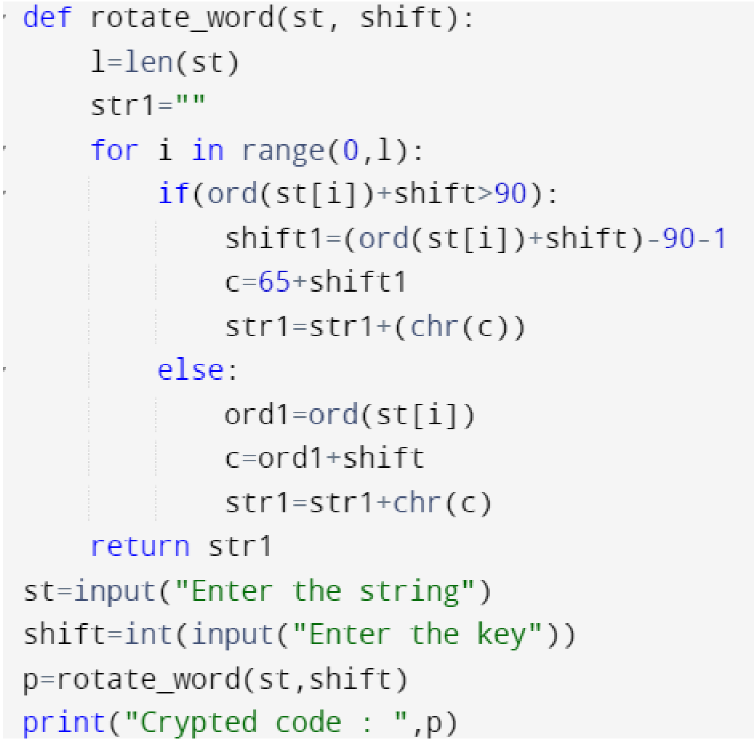
Sample Input:	**RIVER**	**IBM**
Key:	**13**	**-1**
Output:	**EVIRE**	**HAL**

Write a function called rotate_word that takes a string and an integer as parameters, and returns a new string that contains the letters from the original string rotated by the given amount.

Flowchart:

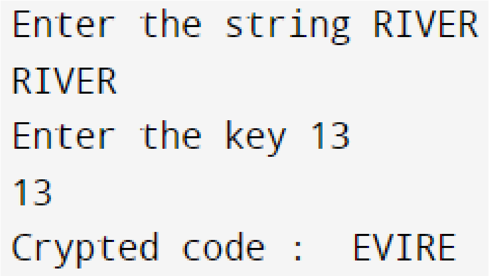

Start
Input st, shift
l= len(str)
str=""
i inrange (0,l-1)?
F
display str1
Stop
T
ord st[i] + shift >90?
F
ord1 = ord (st[i])
c= ord1 + shift
str1 =str1 + chr(c)
T
shift1 = ord(st[i]+ shift)-91
c=65 + shift1
str1= str1 + (chr (c))

Source Code:

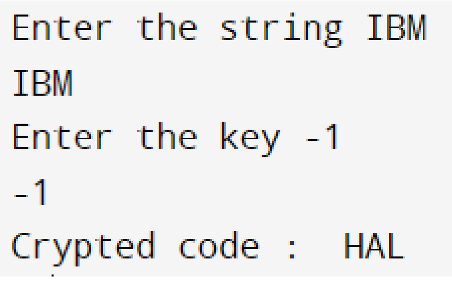


```
def rotate_word(st, shift):
    l=len(st)
    str1=""
    for i in range(0,l):
        if(ord(st[i])+shift>90):
            shift1=(ord(st[i])+shift)-90-1
            c=65+shift1
            str1=str1+(chr(c))
        else:
            ord1=ord(st[i])
            c=ord1+shift
            str1=str1+chr(c)
    return str1
st=input("Enter the string")
shift=int(input("Enter the key"))
p=rotate_word(st,shift)
print("Crypted code : ",p)
```

Output:



```
Enter the string RIVER
RIVER
Enter the key 13
13
Crypted code :  EVIRE
```



```
Enter the string IBM
IBM
Enter the key -1
-1
Crypted code :  HAL
```

Question 2: Write a Python program to convert a given string to Snake case. (Hint: use Join()] Snake case is a variable naming convention where each word is in lower case, and separated by underscores. snake_case refers to the style of writing in which each space is replaced by an underscore (_) character, and all letters of each word written in lowercase. It is a commonly used naming convention in computing, for example for variable and subroutine names, and for filenames. Ignore special characters anywhere in the string, and numbers in the beginning of the string.

For example:

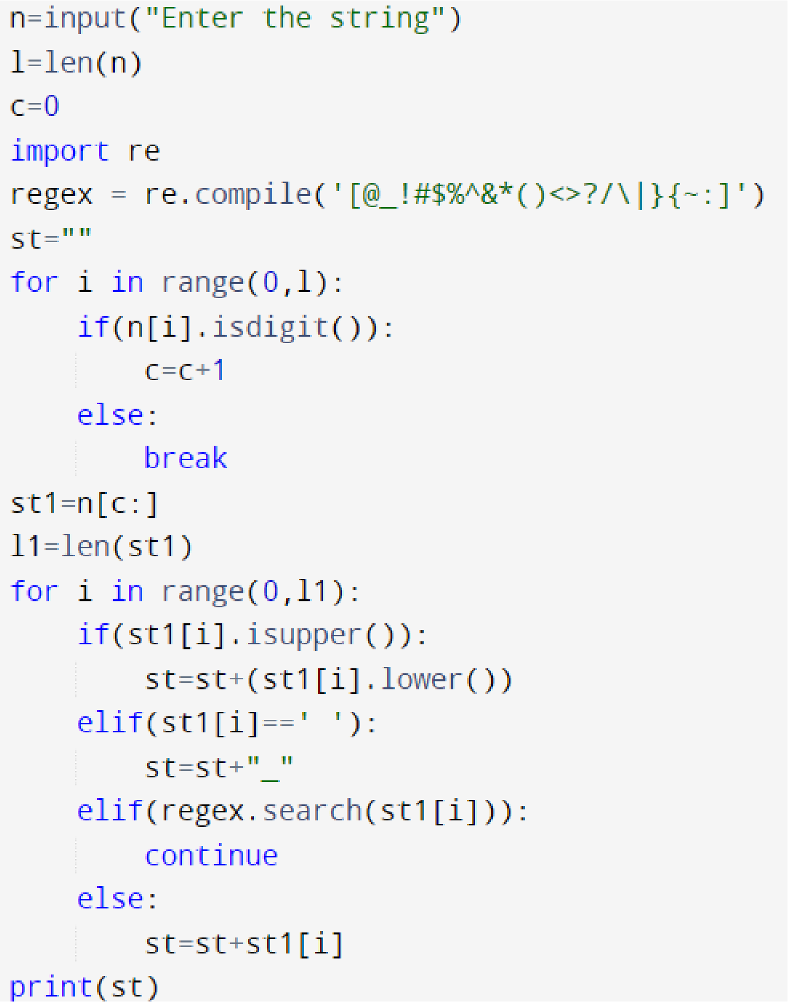
Date of Birth → date_of_birth

This Is In Camel Case → this_is_in_camel_case

123Foo Bar → foo_bar

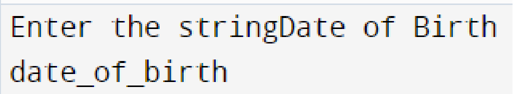
foo3@ Bar# 23 → foo3_bar_23

Source code:

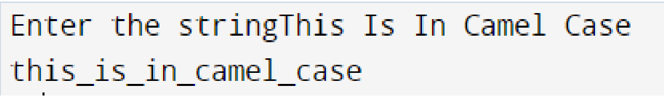


```
n=input("Enter the string")
l=len(n)
c=0
import re
regex = re.compile('[@_!#$%^&*()<>?/\|}{~:]')
st=""
for i in range(0,l):
    if(n[i].isdigit()):
        c=c+1
    else:
        break
st1=n[c:]
l1=len(st1)
for i in range(0,l1):
    if(st1[i].isupper()):
        st=st+(st1[i].lower())
    elif(st1[i]==' '):
        st=st+"_"
    elif(regex.search(st1[i])):
        continue
    else:
        st=st+st1[i]
print(st)
```

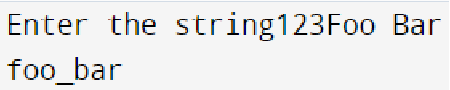
Output:



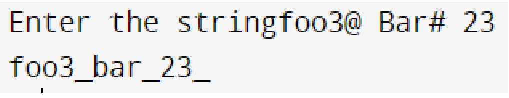
```
Enter the stringDate of Birth
date_of_birth
```



```
Enter the stringThis Is In Camel Case
this_is_in_camel_case
```



```
Enter the string123Foo Bar
foo_bar
```



```
Enter the stringfoo3@ Bar# 23
foo3_bar_23_
```

Question 3: Write a program in python that accepts a string to set up passwords. The program should check the validity of the password. The password is considered valid if it satisfies the following criteria.

A. The password should be at least 8 characters long.

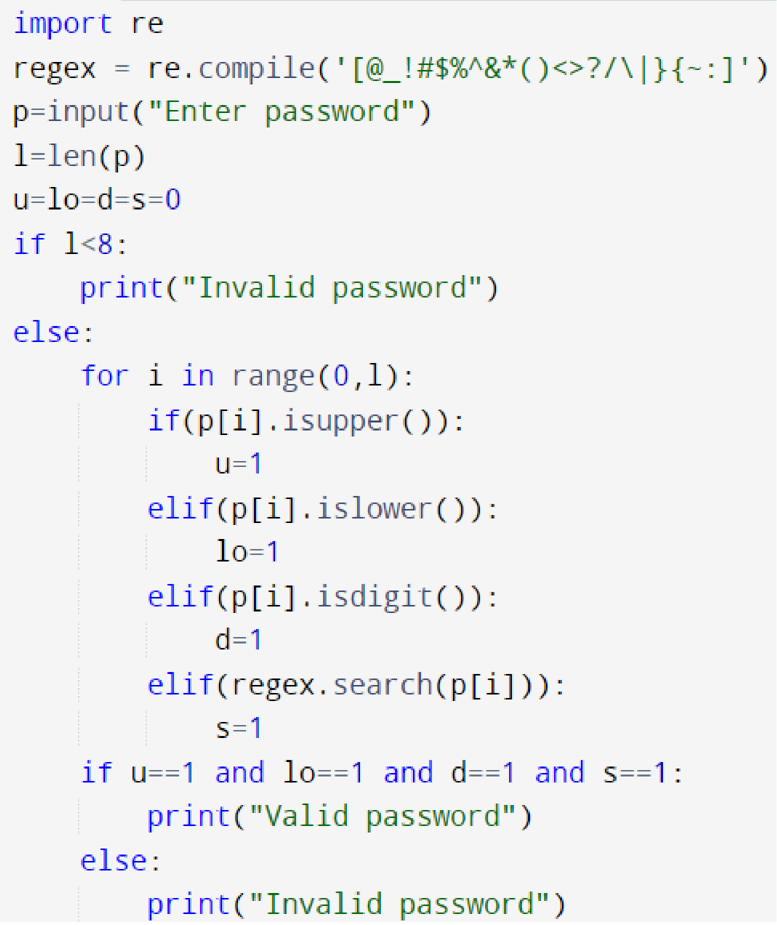
B. The password should contain at least one uppercase letter.

C. The password should contain at least one lowercase letter.

D. The password should contain at least one digit.

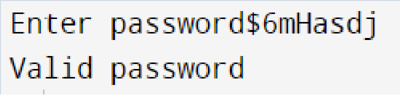
E. The password should contain at least one special character from the following @, #, $, %, &, *, +, -, =, ?, _

Source code:

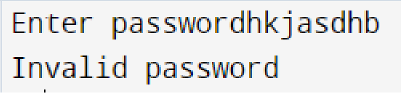


```
import re
regex = re.compile('[@_!#$%^&*()<>?/\|}{~:]')
p=input("Enter password")
l=len(p)
u=lo=d=s=0
if l<8:
    print("Invalid password")
else:
    for i in range(0,l):
        if(p[i].isupper()):
            u=1
        elif(p[i].islower()):
            lo=1
        elif(p[i].isdigit()):
            d=1
        elif(regex.search(p[i])):
            s=1
    if u==1 and lo==1 and d==1 and s==1:
        print("Valid password")
    else:
        print("Invalid password")
```

Output:



```
Enter password$6mHasdj
Valid password
```



```
Enter passwordhkjasdhb
Invalid password
```

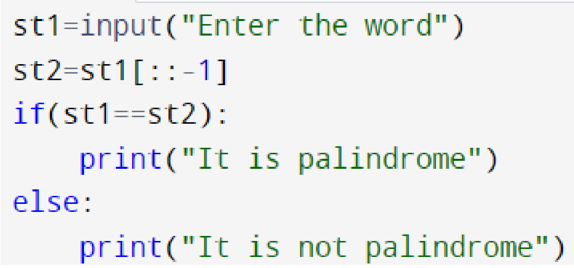
Additional problems using strings for practice:

Question 1: Check whether a string is a palindrome or not. Example: By reversing each letter of REDIVIDER → REDIVIDER detartrated – an 11-letter word that the Guinness Book of World Records says is the longest English palindrome.

Pseudocode:

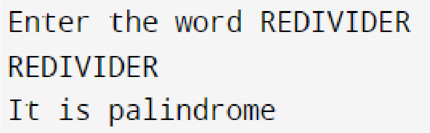
```
BEGIN
READ st1
St2=st1[::-1]
IF(st1==st2):
        DISPLAY "Palindrome"
ELSE:
        DISPLAY "Not palindrome"
```

Source code:

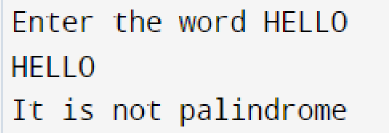


```python
st1=input("Enter the word")
st2=st1[::-1]
if(st1==st2):
    print("It is palindrome")
else:
    print("It is not palindrome")
```

Output:



```
Enter the word REDIVIDER
REDIVIDER
It is palindrome
```



```
Enter the word HELLO
HELLO
It is not palindrome
```

Question 2: Read two strings and find the longest common prefix.

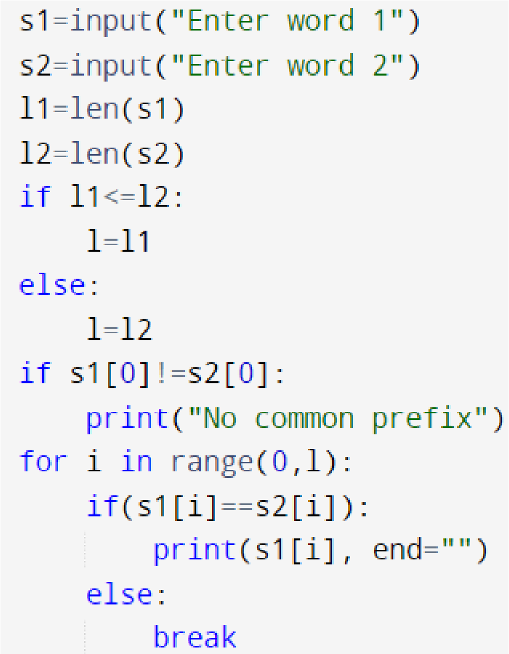
Example:

flower, floor -> flo

pot, plot -> p

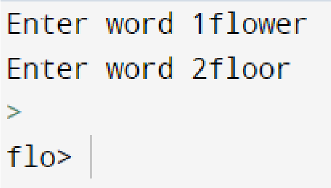
hot, pot -> no common prefix

Source code:

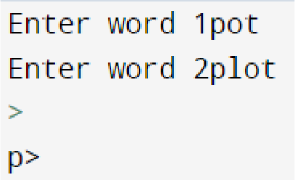


```python
s1=input("Enter word 1")
s2=input("Enter word 2")
l1=len(s1)
l2=len(s2)
if l1<=l2:
    l=l1
else:
    l=l2
if s1[0]!=s2[0]:
    print("No common prefix")
for i in range(0,l):
    if(s1[i]==s2[i]):
        print(s1[i], end="")
    else:
        break
```

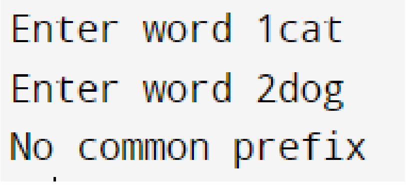
Output:



```
Enter word 1flower
Enter word 2floor
>
flo>
```



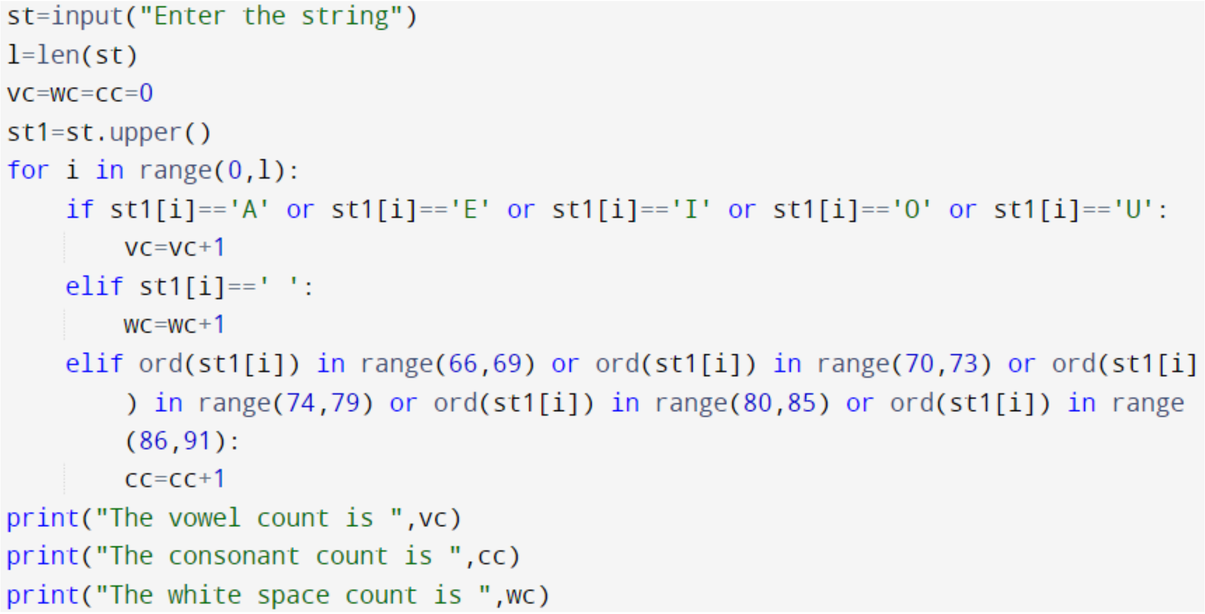
```
Enter word 1pot
Enter word 2plot
>
p>
```



```
Enter word 1cat
Enter word 2dog
No common prefix
```

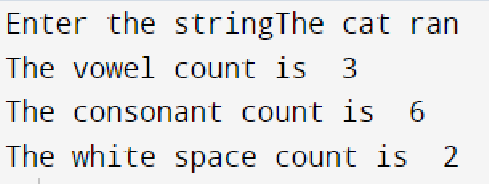
Question 3: Count number of vowels, consonants and white spaces in the string.

Source code:



```python
st=input("Enter the string")
l=len(st)
vc=wc=cc=0
st1=st.upper()
for i in range(0,l):
    if st1[i]=='A' or st1[i]=='E' or st1[i]=='I' or st1[i]=='O' or st1[i]=='U':
        vc=vc+1
    elif st1[i]==' ':
        wc=wc+1
    elif ord(st1[i]) in range(66,69) or ord(st1[i]) in range(70,73) or ord(st1[i]
        ) in range(74,79) or ord(st1[i]) in range(80,85) or ord(st1[i]) in range
        (86,91):
        cc=cc+1
print("The vowel count is ",vc)
print("The consonant count is ",cc)
print("The white space count is ",wc)
```

Output:



```
Enter the stringThe cat ran
The vowel count is  3
The consonant count is  6
The white space count is  2
```

Question 4: Write a menu driven program using inbuilt functions to perform the following:

a. occurrence of a substring.

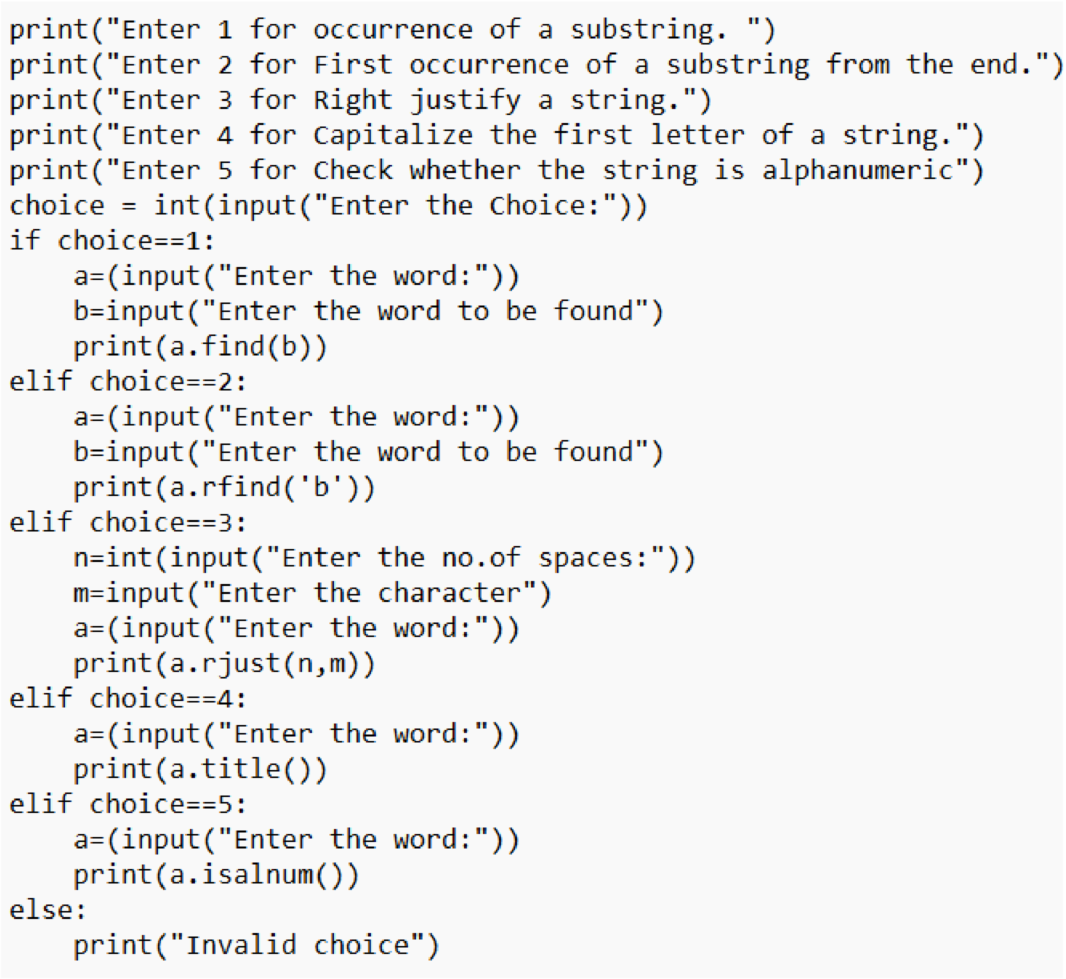
b. First occurrence of a substring from the end.

c. Right justify a string.

d. Capitalize the first letter of a string.

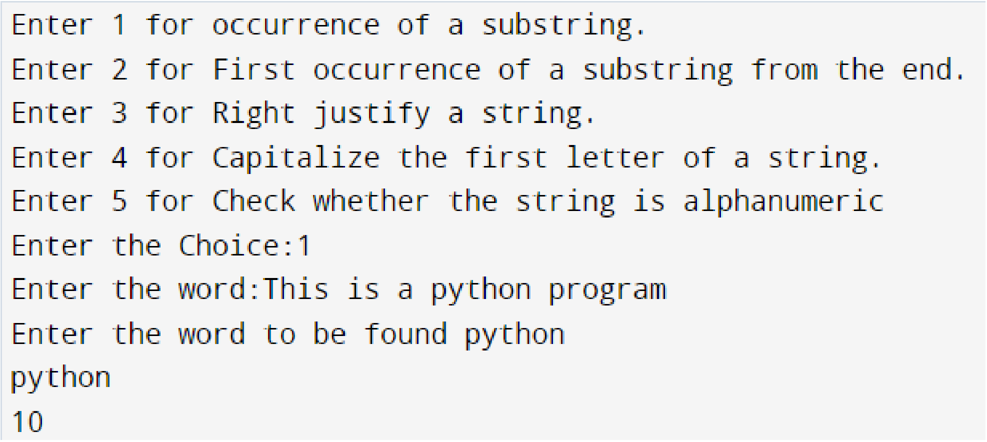
e. Check whether the string is alphanumeric.

Source code:

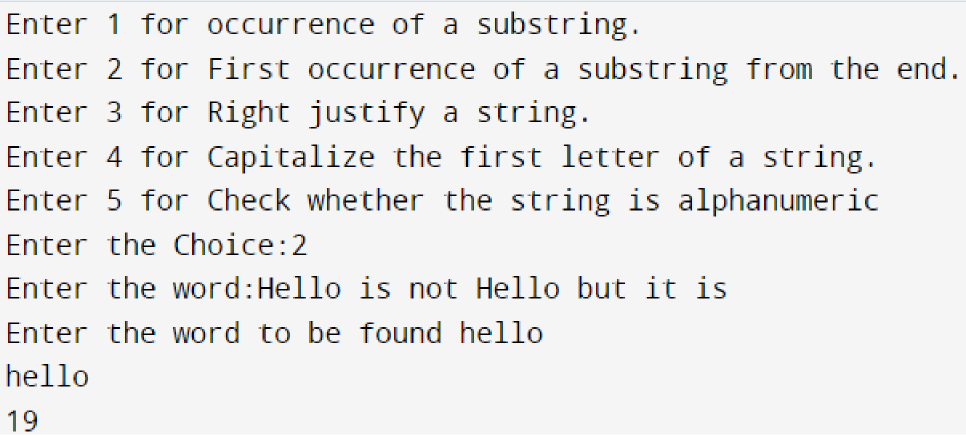


```
print("Enter 1 for occurrence of a substring. ")
print("Enter 2 for First occurrence of a substring from the end.")
print("Enter 3 for Right justify a string.")
print("Enter 4 for Capitalize the first letter of a string.")
print("Enter 5 for Check whether the string is alphanumeric")
choice = int(input("Enter the Choice:"))
if choice==1:
    a=(input("Enter the word:"))
    b=input("Enter the word to be found")
    print(a.find(b))
elif choice==2:
    a=(input("Enter the word:"))
    b=input("Enter the word to be found")
    print(a.rfind('b'))
elif choice==3:
    n=int(input("Enter the no.of spaces:"))
    m=input("Enter the character")
    a=(input("Enter the word:"))
    print(a.rjust(n,m))
elif choice==4:
    a=(input("Enter the word:"))
    print(a.title())
elif choice==5:
    a=(input("Enter the word:"))
    print(a.isalnum())
else:
    print("Invalid choice")
```

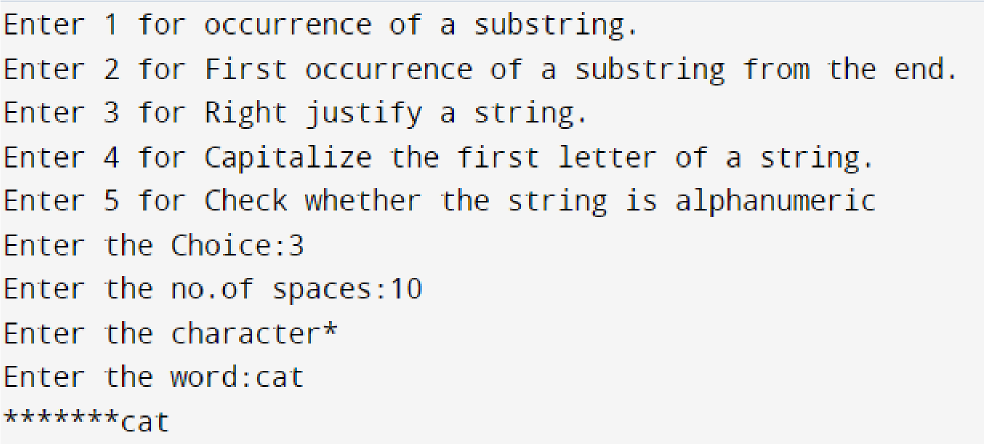
Output:



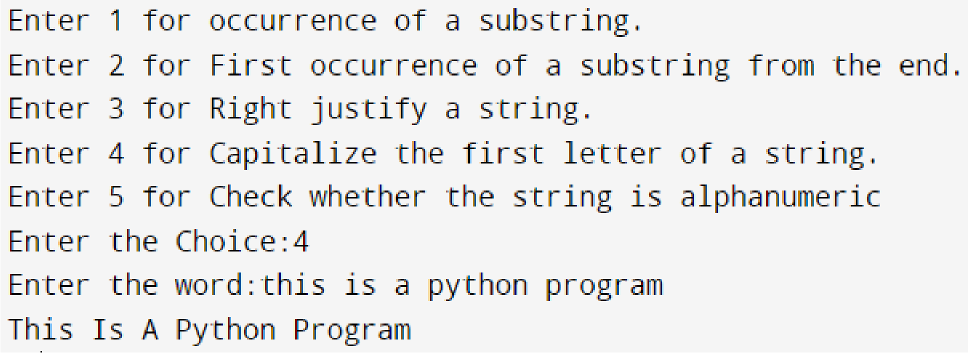
```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:1
Enter the word:This is a python program
Enter the word to be found python
python
10
```



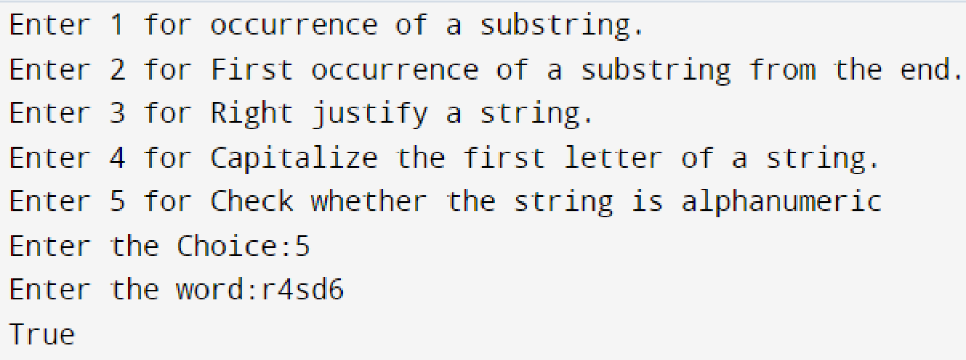
```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:2
Enter the word:Hello is not Hello but it is
Enter the word to be found hello
hello
19
```



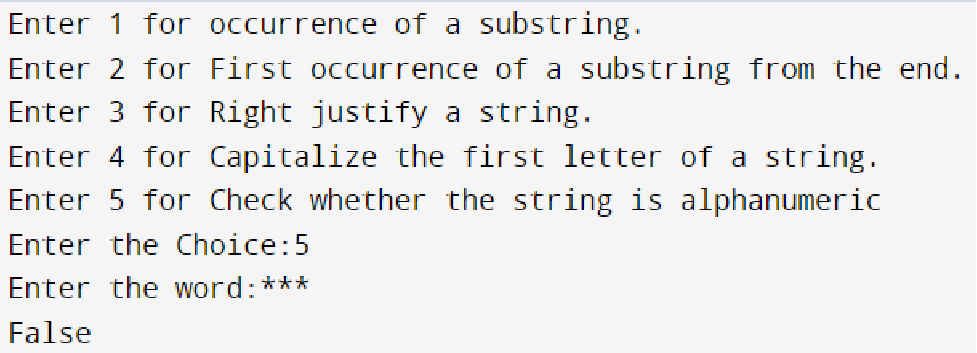
```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:3
Enter the no.of spaces:10
Enter the character*
Enter the word:cat
*******cat
```



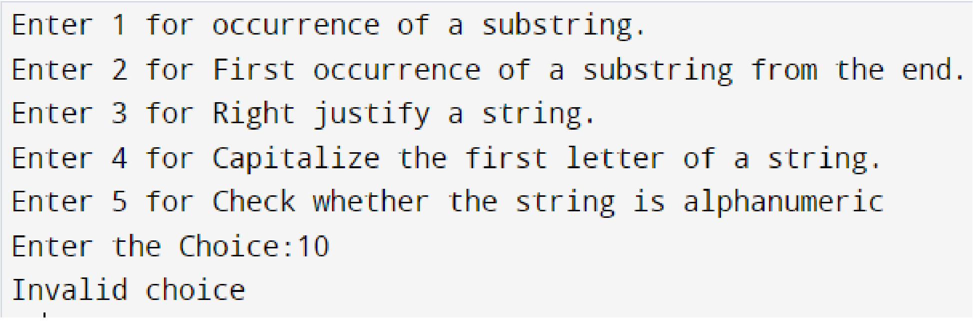
```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:4
Enter the word:this is a python program
This Is A Python Program
```



```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:5
Enter the word:r4sd6
True
```



```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:5
Enter the word:***
False
```



```
Enter 1 for occurrence of a substring.
Enter 2 for First occurrence of a substring from the end.
Enter 3 for Right justify a string.
Enter 4 for Capitalize the first letter of a string.
Enter 5 for Check whether the string is alphanumeric
Enter the Choice:10
Invalid choice
```

Learning outcome:

1. Reading inputs / Printing the result

2. Using appropriate datatypes for the given input

3. Variable assignment

4. Converting the formula into python expressions

Result: Thus I learned to implement a simple problems in Python and solve the same using Strings.