In [ ]:

```
#1. Write a program that asks the user to enter a string. The program should then p
(a) The total number of characters in the string
(b) The string repeated 10 times
(c) The first character of the string (remember that string indices start at 0) (d)
characters of the string
(e) The last three characters of the string
(f) The string backwards
(g) The seventh character of the string if the string is long enough and a message
(h) The string with its first and last characters removed
(i) The string in all caps
(j) The string with every a replaced with an e
```

In [21]:

```
#1. Write a program that asks the user to enter a string. The program should then p
#(a) The total number of characters in the string
string=input("enter a string:")
number=len(string)
print(number)
```

```
49
```

In [22]:

```
#(b) The string repeated 10 times
repeat=string*10
print(repeat)
```

```
Vasanth is an awesome guy i found for my lifetimeVasanth is an awesome guy i found f
or my lifetimeVasanth is an awesome guy i found for my lifetimeVasanth is an awesome
guy i found for my lifetimeVasanth is an awesome guy i found for my lifetimeVasanth
is an awesome guy i found for my lifetimeVasanth is an awesome guy i found for my li
fetimeVasanth is an awesome guy i found for my lifetimeVasanth is an awesome guy i f
ound for my lifetimeVasanth is an awesome guy i found for my lifetime
```

In [ ]:

```
#(c) The first character of the string (remember that string indices start at 0) (d
#characters of the string
```

In [23]:

```
#method-1:
print(string[0])
```

```
V
```

In [24]:

```
#method-2:
words=string.split()
print(words[0])
```

```
Vasanth
```

In [ ]:

```
#(d) The first three characters of the string
```

In [25]:

```
#method-1:
print(string[0:3])
```

```
Vas
```

In [ ]:

```
#(e) The last three characters of the string
```

In [26]:

```
print(string[len(string)-3:])
```

```
ime
```

In [ ]:
```
#(f) The string backwards
```

In [28]:
```
print(string[::-1])
```

```
emitefil ym rof dnuof i yug emosewa na si htnasaV
```

In [ ]:
```
#(g) The seventh character of the string if the string is long enough and a message
```

In [34]:
```
if len(string)>=7:
    print(string[6])
else:
    print("string is not long enough")
```

```
h
```

In [ ]:
```
#(h) The string with its first and last characters removed
```

In [31]:
```
string[1:len(string)-1]
```

Out[31]:
```
'asanth is an awesome guy i found for my lifetim'
```

In [ ]:
```
#(i) The string in all caps
```

In [32]:
```
string.upper()
```

Out[32]:
```
'VASANTH IS AN AWESOME GUY I FOUND FOR MY LIFETIME'
```

In [ ]:
```
#(j) The string with every a replaced with an e
```

In [33]:
```
string.replace('a','e')
```

Out[33]:
```
'Vesenth is en ewesome guy i found for my lifetime'
```

In [ ]:
```
#2. A simple way to estimate the number of words in a string is to count the number
string. Write a program that asks the user for a string and returns an estimate of
are in the string.
Tip: You need to count the number of words using spaces
```

In [38]:
```
string=input("enter string:")
number_words=string.count(' ')+1
print(number_words)
```

```
10
```

In [ ]:
```
#3.Write a program that asks the user to enter a word and prints out whether that w
vowels.
```

In [18]:
```
word=input("enter a word:")
vowels = "aeiouAEIOU"
for char in word:
    if char in vowels:
        print('yes')
        break
```

```
yes
```

In [19]:

```
word=input("enter a word:")
vowels = "aeiouAEIOU"
for char in word:
    if char in vowels:
        print('yes')
        break
```

In [ ]:

```
#4. Improvise above code by providing unique vowels
```

In [ ]:

In [ ]:

```
#5. Write a program that asks the user to enter a string. The program should create
new_string from the user’s string such that the second character is changed to an a
exclamation points are attached to the end of the string. Finally, print new_string
Typical output is shown below:
Enter your string: Qbert
Output: Q*ert!!!
```

In [25]:

```
string=input("enter your string:")
new_string=' '
s=string.replace('b','*')
s1='!!!'
print(s+s1)
```

```
Q*ert!!!
```

In [ ]:

```
#6. Write a program that asks the user to enter a word and determines whether the w
palindrome or not. A palindrome is a word that reads the same backwards as forwards
```

In [32]:

```
word=input("enter a word:").lower()
reverse=word[::-1].lower()
if word==reverse:
    print('polindrome')
else:
    print('not a polindrome')
```

```
polindrome
```

In [33]:

```
word=input("enter a word:")
reverse=word[::-1]
if word==reverse:
    print('polindrome')
else:
    print('not a polindrome')
```

```
not a polindrome
```

In [ ]:

```
#7. At a certain school, student email addresses end with @student.college.edu, whi
email addresses end with @prof.college.edu. Write a program that first asks the use
email addresses they will be entering, and then has the user enter those addresses.
email addresses are entered, the program should print out a message indicating eith
addresses are student addresses or that there were some professor addresses entered
```

In [116…

```
num_addresses=eval(input("enter how many email addresses you want to enter:"))
count_studentemail=0
count_professoremail=0
for i in range(num_addresses):
    email=input(f"enter email id:{i+1}")
    if email.endswith('@student.college.edu'):
        count_studentemail=count_studentemail+1
    elif email.endswith('@prof.college.edu'):
        count_professoremail=count_professoremail+1
if count_studentemail==num_addresses:
    print("all the addresses are student addresses")
elif count_professoremail==num_addresses:
    print("all the addresses are professor addresses")
else:
    print("there were some professor addresses entered")
```

```
there were some professor addresses entered
```

In [ ]:

```
#8. Write a program that asks the user to enter a string, then prints out each Lett
doubled and on a separate line. For instance,
if the user entered HEY,
the output would be
HH
EE
YY
```

In [37]:

```
string=input("enter a string:")
for i in range(len(string)):
    print(string[i]*2)
```

```
HH
EE
YY
```

In [ ]:

```
#9.Write a program that asks the user to enter a word that contains the letter a. T
then print the following two lines: On the first line should be the part of the str
including the the first a, and on the second line should be the rest of the string.
Sample output is shown below:
Enter a word: buffalo
buffa
lo
```

In [46]:

```
word=input("enter a word:")
first=word.index('a')
before_a=word[:first+1]
after_a=word[first+1:]

print(before_a)
print(after_a)
```

```
dora
emon
```

In [15]:

```
#10. Write a program that asks the user to enter a word and then capitalizes every
So if the user enters rhinoceros,
the program should print rHiNoCeRoS.
```

In [104…

```
word=input("enter a word:")
list=[]
for i in range(len(word)):
    if i %2==0:
        list.append(word[i].upper())
    else:
        list.append(word[i].lower())
''.join(list)
```

Out[104…

```
'rHiNoCeRoS'
```

In [ ]:

```
#11. Write a program that asks the user to enter two strings of the same length. Th
then check to see if the strings are of the same length. If they are not, the progr
appropriate message and exit. If they are of the same length, the program should al
characters of the two strings. For example,
if the user enters abcde and ABCDE
the program should print out AaBbCcDdEe
```

In [26]:

```
str1=input("enter first string:")
str2=input("enter second string:")
list=[]
if len(str1)!=len(str2):
    print("the string are of not same length")
else:
    for i in range(len(str1)):
        list.append(str2[i])
        list.append(str1[i])
        a="".join(list)
    print(a)
```

```
the string are of not same length
```

In [28]:

```
str1=input("enter first string:")
str2=input("enter second string:")
list=[]
if len(str1)!=len(str2):
    print("the string are of not same length")
else:
    for i in range(len(str1)):
        list.append(str2[i])
        list.append(str1[i])
        a="".join(list)
    print(a)
```

```
VSahSRauNTthHI
```

In [ ]:

```
#12. Write a program that asks the user to enter their name in lowercase and then c
letter of each word of their name
```

In [33]:

```
name=input("enter name in lowercase:")
name.title()
```

Out[33]:

```
'Shruthi Vangari'
```

In [ ]:

```
#13. The goal of this exercise is to see if you can mimic the behavior of the in op
and index methods using only variables, for loops, and if statements.
(a) Without using the in operator, write a program that asks the user for a string
and prints out whether or not the letter appears in the string.
(b) Without using the count method, write a program that asks the user for a string
letter and counts how many occurrences there are of the letter in the string.
(c) Without using the index method, write a program that asks the user for a string
letter and prints out the index of the first occurrence of the letter in the string
not in the string, the program should say so.
```

In [ ]:

```
#13. The goal of this exercise is to see if you can mimic the behavior of the in op
and index methods using only variables, for loops, and if statements.
(a) Without using the in operator, write a program that asks the user for a string
and prints out whether or not the letter appears in the string
```

In [42]:

```
string=input("enter a string:")
letter=input("enter a letter:")
found=False
for i in string:
    if i==letter:
        found=True
        break
if found:
    print(f"the {letter} appears in string")
else:
    print(f"the {letter} not appear in string")
```

```
the v appears in string
```

In [ ]:

```
#(b) Without using the count method, write a program that asks the user for a strin
letter and counts how many occurrences there are of the letter in the string.
```

In [46]:

```
string=input("enter a string:")
letter=input("enter a letter:")
count=0
for i in string:
    if i==letter:
        count=count+1
print(count)
```

```
10
```

In [ ]:

```
#(c) Without using the index method, write a program that asks the user for a strin
letter and prints out the index of the first occurrence of the letter in the string
not in the string, the program should say so.
```

In [56]:

```
string=input("enter a string:")
letter=input("enter a letter:")
ind=-1
for i in range(len(string)):
    if string[i]==letter:
        ind=i
        break
```

```
if ind!=-1:
    print(f"index of first occurrence of letter is {ind}")
else:
    print(f"letter is not present in the string")
```

```
index of first occurrence of letter is 1
```

In [ ]:

```
#14. Finding a substring within a string
For example, if we were presented a series of lines formatted as follows:
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
and we wanted to pull out only the second half of the address (i.e., uct.ac.za
```

In [96]:

```
string='From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008'
i1=string.find('@')
i1
i2=string.find(' ')
i2
i3=string.find(' ',i2+1)
i3
s=string[i1+1:i3+1]
print(s)
```

```
uct.ac.za
```

In [ ]:

```
#15. Write a Python program to add 'ing' at the end of a given string (length shoul
given string already ends with 'ing' then add 'ly' instead.
If the string length of the given string is less than 3, leave it unchanged.
Go to the editor
Sample String : 'abc'
Expected Result : 'abcing'
Sample String : 'string'
Expected Result : 'stringly'
```

In [73]:

```
string=input("enter a string:")
if string.endswith('ing'):
    new_string=string +'ly'
    print(new_string)
elif len(string)>=3:
    new_string=string+'ing'
    print(new_string)
elif len(string)<3:
    print(string)
```

```
stringly
```

In [74]:

```
string=input("enter a string:")
if string.endswith('ing'):
    new_string=string +'ly'
    print(new_string)
elif len(string)>=3:
    new_string=string+'ing'
    print(new_string)
elif len(string)<3:
    print(string)
```

```
abcingly
```

In [75]:

```
string=input("enter a string:")
if string.endswith('ing'):
    new_string=string +'ly'
    print(new_string)
elif len(string)>=3:
    new_string=string+'ing'
    print(new_string)
elif len(string)<3:
    print(string)
```

```
HE
```

In [ ]:

```
#16. Take the following Python code that stores a string:‘
string = 'X-DSPAM-Confidence: 0.8475'
Extract the portion of the string after the colon character and then use the float
the extracted string into a floating point number
```

In [89]:

```
string = 'X-DSPAM-Confidence: 0.8475'
i1=string.find(':')
i1
substring=string[i1+1:]
trimm=substring.strip()
floatt=float(trimm)
print(floatt)
```

```
0.8475
```

In [ ]: