In [ ]:

```
# 1. Write a program that asks the user to enter a string. The program sho
# (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 star
# (d) The first three 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
# (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 [62]:

```
str1=input("Enter a String: \n")
a=print('a) The total number of Charaters in a String is \n',len(str1))
print('b) The string is printed below 10 times \n')
for i in range(1,11):
    print(str1)
print('(c) The first character of the string \n',str1[0])
print('(d) The first three characters of the string \n ',str1[0],str1[1],s
print('(e) The last three characters of the string \n',str1[-1],str1[-2],s
print('(f) The string backwards :\n')
for i in reversed(str1):
    print("\n",i,end=' ')
print('(g) The seventh character of the string if the string is long enoug
if len(str1)>7:
    print(" The seventh charcater of the String is \n",str1[8])
else:
    print('The string is very small to return the value \n')
print('(h) The string with its first and last characters removed: \n')
for i in range(1,len(str1)-1):
    print(str1[i],end=' ')

print('(i) The string in all caps \n',str1.upper())
print('(j) The string with every" a" replaced with an "e" \n',str1.replace
```

```
Enter a String:
Aishwarya
a) The total number of Charaters in a String is
 9
b) The string is printed below 10 times

Aishwarya
Aishwarya
Aishwarya
Aishwarya
Aishwarya
Aishwarya
Aishwarya
Aishwarya
Aishwarya
Aishwarya
(c) The first character of the string
 A
(d) The first three characters of the string
  A i s
(e) The last three characters of the string
 a y r
(f) The string backwards :


 a
 y
 r
 a
 w
 h
 s
 i
 A (g) The seventh character of the string if the string is long enough
and a message otherwise

 The seventh charcater of the String is
 a
(h) The string with its first and last characters removed:

i s h w a r y (i) The string in all caps
 AISHWARYA
(j) The string with every" a" replaced with an "e"
 Aishwerye
```

In [ ]:

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

In [130]:
```
count =1
str2=input('Enter the string : ')
for i in str2:
    if i == ' ':
        count+=1
print('The number of words',count)
```

```
Enter the string : HI MY NAME IS VIDYA
The number of words 5
```

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

In [113]:
```
str3=input('Enter the String:')
for i in str3:
    if i in ('a','e','i','o','u'):
        print(' vowels  are',i)
    else:
        print('The letters which are not vowels are',i)
```

```
Enter the String:aishwraya
 vowels  are a
 vowels  are i
The letters which are not vowels are s
The letters which are not vowels are h
The letters which are not vowels are w
The letters which are not vowels are r
 vowels  are a
The letters which are not vowels are y
 vowels  are a
```

In [116]:
```
# 4. Improvise above code by providing unique vowels
# str4=input('Enter the String:')
# for i in str4:
#     if i in ('a','e','i','o','u'):
#         print(' vowels  are',i)
#     else:
#         print('The letters which are not vowels are',i)

str4=input('Enter the String:')
str44=""
p=len(str4)
for i in str4:
    if i in 'aeiou':
        if i not in str44:
            str44=str44+i


print(f"The number of ovwels are {len(str44)}")
```

```
Enter the String:we are herew
The number of ovwels are 2
```

In [121]:
```
# 5. Write a program that asks the user to enter a string.
# The program should create a new string called new_string from the user's
# string such that the second character is changed to an asterisk and thre
# exclamation points are attached to the end of the string. Finally, print
# Typical output is shown below:
# Enter your string: Qbert
# Output:  Q*ert!!!


str5=input('Enter a String:')
str55=''
i1=str5.replace(str5[1],'*')
print(i1,end='!!!')
```

```
Enter a String:Aishwarya
A*shwarya!!!
```

In [124]:
```
# 6. Write a program that asks the user to enter a word and determines whe
# is a palindrome or not. A palindrome is a word that reads the same backw

str6=input('Enter a String:')
p=str6[::-1]
p
if str6==p:
    print('The given input is a palindrome')
else:
    print('The given input is not a plaindrome')
```

```
Enter a String:wsedrftg
The given input is not a plaindrome
```

In [42]:
```
# 7. At a certain school, student email addresses end with @student.colleg
# while professor email addresses end with @prof.college.edu. Write a prog
# first asks the user how many email addresses they will be entering, and
# enter those addresses. After all the email addresses are entered, the pr
# message indicating either that all the addresses are student addresses o
# addresses entered.

# student email addresses end with @student.college.edu
# professor email addresses end with @prof.college.edu
count1=0
count2=0
stu_email='@student.college.edu'
prof_email='@prof.college.edu'
str7=eval(input('How many email address will be entered'))
for i in range(0,str7):
    sdf=input('Enter the email:')
    if sdf==stu_email:
        print('Its student email')
        count1+=1
    else:
        print('Its professsor email')
        count2+=1
print('The number of student mail is',count1)
print('The number of professsor mail is',count2)
```

```
How many email address will be entered2
Enter the email:@student.college.edu
Its student email
Enter the email:@student.college.edu
Its student email
The number of student mail is 2
The number of professsor mail is 0
```

In [128]:

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

str8=input('Enter the String:')
for i in str8:
    print(i,i)
```

```
Enter the String:HEY
H H
E E
Y Y
```

In [117]:
```
dir('')
```

Out[117]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'capitalize',
 'casefold',
 'center',
 'count',
 'encode',
 'endswith',
 'expandtabs',
 'find',
 'format',
 'format_map',
 'index',
 'isalnum',
 'isalpha',
 'isascii',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
 'istitle',
 'isupper',
 'join',
```

```
 'ljust',
 'lower',
 'lstrip',
 'maketrans',
 'partition',
 'removeprefix',
 'removesuffix',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [21]:

```
# 9.Write a program that asks the user to enter a word that contains the l
# The program should then print the following two lines: On the first line
# and including the the first a, and on the second line should be the rest
# Sample output is shown below:
# Enter a word: buffalo
# buffa
# lo
str9=input('Enter the String:')
m=str9.index('a')
n=len(str9)

print(str9[0:m+1])
print(str9[m+1:n])
```

```
Enter the String:buffalo
buffa
lo
```

In [25]:
```
# 10. Write a program that asks the user to enter a word and then capitali
# So if the user enters rhinoceros,
# the program should print rHiNoCeRoS.
output=''
str10=input('Enter a string: ')
for i in range( len(str10)):
    if i%2==0:
        output+=str10[i].lower()
    else:
        output+=str10[i].upper()

print(output)
```

```
Enter a string: qwertyuiop
qWeRtYuIoP
```

In [ ]:
```
# 11. Write a program that asks the user to enter two strings of the same
# The program should then check to see if the strings are of the same leng
# If they are not, the program should print an appropriate message and exi
# the program should alternate the characters of the two strings. For exam
# if the user enters abcde and ABCDE
# the program should print out AaBbCcDdEe.
str111=input('Enter the string of certain length: ')
str112=input('Enteer the string of similar length')
if len(str111)==len(str112):
    print("The length is similar hence the program continues:")

else:
    print("The length of the strings are different hence the program exits
```

In [11]:
```
# 12. Write a program that asks the user to enter their name in lowercase
# then capitalizes the first letter of each word of their name.
str12=input('Enter the string in lowercase: ')
m=str12.title()
print(m)
```

```
Enter the string: i am the happiest person in this world
I Am The Happiest Person In This World
```

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

In [5]:
```
# (a) Without using the in operator, write a program that asks the user fo
# letter and prints out whether or not the letter appears in the string.
str131=input('Enter a string: ')
str1311=input('Enter a character form a string whose exisitace in string w
if str1311 in str131:
    print('The letter exists in the word')
else:
    print('The letter doesnt exist in the word')
```

```
Enter a string: awsderf
Enter a character form a string whose exisitace in string we are suppose
d to check: r
The letter exists in the word
```

In [9]:
```
# (b) Without using the count method, write a program that asks the user f
# a letter and counts how many occurrences there are of the letter in the
count=0
str132=input('Enter a string: ')
str1312=input('Enter a character form a string whose exisitace in string w
for i in str132:
    if str1312 in str132:

        count+=1
    else:

        count+=0
print('The number of times the letter was repeated is',count)
```

```
Enter a string: asihwatya
Enter a character form a string whose exisitace in string we are suppose
d to check: a
The number of times the letter was repeated is 10
```

In [27]:
```
# (c) Without using the index method, write a program that asks the user f
# a letter and prints out the index of the first occurrence of the letter
# is not in the string, the program should say so.
try:
    str133=input('Enter a string: ')
    str1313=input('Enter a character form a string whose exisitace in stri
    for i in str132:
        if str1312 in str132:

            count+=1
            print(count[i])
        else:

            count+=0
   # print('The number of times the letter was repeated is',str1313[i])
except Exception as e:
    print(e)
```

```
Enter a string: qwe
Enter a character form a string whose exisitace in string we are suppose
d to check: q
'int' object is not subscriptable
```

In [9]:
```
# 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
str14 = 'stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008'
m=str14.index('@')
n=str14.index('za')
print('The second half of the address is',str14[m:n+2])
```

```
The second half of the address is @uct.ac.za
```

In [6]:
```
# 15.Write a Python program to add 'ing' at the end of a given string (len
# If the given string already ends with 'ing' then add 'ly' instead.
# If the string length of the given string is less than 3, leave it unchan
# Go to the editor
# Sample String : 'abc'
# Expected Result : 'abcing'
# Sample String : 'string'
# Expected Result : 'stringly'

str15=input("Enter the string: ")
if len(str15)>3:

    m=str15.endswith('ing')
    if m==True:

        print(str15 + 'ly')
    else:
        print(str15 + 'ing')
else:
    print('As the string is less than 3 we are returning the value', str15
```

```
Enter the string: wuhn
wuhning
```

In [36]:
```
# 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 th
#     convert the extracted string into a floating point number.

str16='X-DSPAM-Confidence: 0.8475'
m=str16.index(':')
p=str16[m+1:len(str16)]
l=float(p)
print('The floating point number is', l)
print('The datatype is',type(l))
```

```
The floating point number is 0.8475
The datatype is <class 'float'>
```

In [ ]: