LAB CYCLE - 2

Experiment No :1

Date :17/10/2024

Aim :

Create a String from the given string where the first and last character are exchanged.

Pseudocode :

```
DISPLAY "Enter a string: "
GET str
SET newstring = str[-1]+str[1:-1]+str[0]
DISPLAY "Old string :",str
DISPLAY "newstring: ",newstring
```

Souce Code :

```
str=input("Enter a string\n")
newstring=str[-1]+str[1:-1]+str[0]
print("Old string:",str)
print("newstring:",newstring)
```

Output :

```
Enter a string
College
Old string: college
Newstring: eollegc
```

Result : The program is successfully executed and the output is verified.

Experiment No :2

Date: 17/10/2024

Aim :

Get a String from the input string where all occurrences of the first Character are replaced with “$”, except first character.

Pseudocode :

```
DISPLAY "Enter a string: "
GET str
SET d = "$"
SET newstring = str[0]+str[1:].replace(str[0],d)
DISPLAY "newstring: ",newstring
```

Method :

Functions	Description	Syntax
str.replace()	It replaces repeated string with another string	str.replace(old,new)

Souce Code :

```
str=input("Enter a string\n").lower()
d="$"
newstring=str[0]+str[1:].replace(str[0],d)
print("newstring:",newstring)
```

Output :

```
Enter a string
Ananthan
newstring:An$nth$n
```

Result : The program is successfully executed and the output is verified.

Experiment No :3

Date: 17/10/2024

Aim :

Create a single string separated by space from 2 strings by swapping the characters at position 1.

Pseudocode :

```
DISPLAY "Enter the first string "
GET strone
DISPLAY "Enter the second string: "
GET strtwo

SET newstr1 = strone[0]+strtwo[1:2]+strone[2:]
SET newstr2 = strtwo[0]+strone[1:2]+strtwo[2:]

DISPLAY newstr1+" "+newstr2
```

Souce Code :

```
strone=input("Enter the first string\n")
strtwo=input("Enter the second string\n")
newstr1=strone[0]+strtwo[1:2]+strone[2:]
newstr2=strtwo[0]+strone[1:2]+strtwo[2:]
print(newstr1+" "+newstr2)
```

Output :

```
Enter the first string
hello
Enter the second string
world
hollo werld
```

Result : The program is successfully executed and the output is verified.

Experiment No :4

Date: 17/10/2024

Aim :

Count the number of characters in a string.

Pseudocode :

```
DISPLAY "Enter the string: "
GET str

SET char_count = empty dictionary

FOR each character i in str DO
    IF i is in char_count THEN
        Increment char_count[i] by 1
    ELSE
        SET char_count[i] = 1
DISPLAY " Character frequency: "
DISPLAY char_count
```

Souce Code :

```
str=input("Enter the string: ")
char_count={}
for char in str:
        if char in char_count:
                char_count[char]+=1
        else:
                char_count[char]=1
print("Character frequency: ")
print(char_count)
```

Output :

```
Enter the string: malayalam
Character frequency:
{‘m’: 2, ‘a’: 4, ‘l’: 2, ‘y’: 1}
```

Result : The program is successfully executed and the output is verified.

Experiment No :5

Date: 17/10/2024

Aim :

Add ‘ing’ at the end of the given string, if it ends with ‘ing’ the add ‘ly’

Pseudocode :

```
DISPLAY "Enter a word: "
GET str
IF last 3 characters of str are "ing" THEN
    DISPLAY str + "ly"
ELSE
    DISPLAY str + "ing"
```

Method :

Functions	Description	Syntax
str.endswith()	method is used to check if a string **ends** with a specified suffix.	str.endswith(“string”)

Souce Code :

```
str=input("Enter a word: \n")
if(str[-3:])=="ing":
        print(str+"ly")
else:
        print(str+"ing")
```

Output :

```
Enter a word:
act
acting

Enter a word:
acting
actingly
```

Result : The program is successfully executed and the output is verified.

Experiment No :6

Date: 17/10/2024

Aim :

Create and Store a list of first names. Count the occurrences of ‘a’ within the list.

Pseudocode :

```
SET name_list=[]
DISPLAY " Enter the number of names:    "
GET n

FOR each character I in range(n) DO
        SET name=input("Enter a first name:").lower()
        SET name_list.append(name)
        SET count_a=0
FOR each character name in name_list
        SET count_a+=name.count('a')

DISPLAY "Total occurence of 'a' and 'A': ",count_a
```

Method :

Functions	Description	Syntax
str.lower()	It is used to convert the string from uppercase to lowercase	name=ABC name.lower()=abc
str.count()	method in Python is used to count the occurrences of a specific value in a sequence of string	sequence.count(value)

Souce Code :

```
name_list=[]
n=int(input("Enter the number of names: "))
for i in range(n):
          name=input("Enter a first name:").lower()
          name_list.append(name)
          count_a=0
for name in name_list:
          count_a+=name.count('a')
print("Total occurence of 'a' and 'A': ",count_a)
```

Output :

```
Enter the number of names:    4
Enter a first name:amal
Enter a first name:ashin
Enter a first name:ananthan
Enter a first name:amrita
Total occurrence of ‘a’ and ‘A’:    8
```

Result : The Program is successfully executed and the output is verified.

Experiment No :7

Date: 17/10/2024

Aim : Write a program to read 2 lists color list 1 and color list 2 print the all the colors from the color list 1 and not contained in color list 2

Pseudocode :

```
DISPLAY "Enter color for list1 seperated by comma:"
GET color_list1
SPLIT color_list1 by ','

DISPLAY "Enter color for list2 seperated by comma:"
GET color_list2
SPLIT color_list2 by ','

SET color_list1=set(color_list1)
SET color_list2=set(color_list2)
SET result=color_list1-color_list2

DISPLAY " colors in list 1 but not in list 2 are: ",result
```

Method :

Functions	Description	Syntax

str.split()	method in Python is used to divide a string into a list of substrings based on a specified delimiter (separator).	str1=a,b,c
		str1.split(‘,’)

Souce Code :

```
color_list1=input("Enter color for list1 seperated by comma:" ).split(",")
color_list2=input("Enter color for list2 seperated by comma:" ).split(",")
color_list1=set(color_list1)
color_list2=set(color_list2)
result=color_list1-color_list2
print("colors in list1 but not in list2 are: ",result)
```

Output :

```
Enter color for list1 seperated by comma:red,yellow,green
Enter color for list2 seperated by comma:black,yellow,blue
```

```
colors in list1 but not in list2 are: {‘green’, ‘red’}
```

Result : The program is successfully executed and the output is verified.

Experiment No :8

Date: 17/10/2024

Aim :

Create a list of colors from comma-separated color names entered by the user. Display first and last colors.

Pseudocode :

```
DISPLAY "Enter a list of colors seperated by commas: "
GET color
SPLIT color by ','

DISPLAY "First color: ",colors[0]
DISPLAY "last color: ",colors[-1]
```

Souce Code :

```
colors=input("Enter a list of colors seperated by commas: ").split(",")
print("First color: ",colors[0])
print("last color: ",colors[-1])
```

Output :

```
Enter a list of colors seperated by commas: red,yellow,black,white
first color: red
last color: white
```

Result : The program is successfully executed and the output is verified.

Experiment No :9

Date: 17/10/2024

Aim :

Write a program to prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

Pseudocode :

```
DISPLAY "Enter integers separated by comma: "
GET a
SPLIT a by ','

SET result = empty list

FOR each num in a DO
    IF integer value of num > 100 THEN
        APPEND "over" to result
    ELSE
        APPEND integer value of num to result

DISPLAY result
```

Method :

Functions	Description	Syntax
str.append()	method is used to add a single item to the end of a list. It modifies the original list in place and does not return a new list.	list.append(item)

Souce Code :

```
a=input("Enter integers seperated by comma: ").split(",")
result=[]
for num in a:
        if int(num)>100:
                result.append('over')
        else:
                result.append(num)
print(result)
```

Output :

```
Enter integers separated by comma:100,250,33,26,170
[100, 'over' , 33 , 26 , 'over']
```

Result : The program is successfully executed and the output is verified.

Experiment No :10

Date: 17/10/2024

Aim :

From a list of integers, create a list after removing even numbers.

Pseudocode :

```
DISPLAY "Enter integers separated by comma: "
GET str
SPLIT str by ','

SET result = empty list

FOR each num in str DO
    SET number=int(num)
    IF integer value of num is odd THEN
        APPEND num to res

DISPLAY (result)
```

Souce Code :

```
str=input("Enter integers seperated by comma: ").split(",")
result=[]
for num in str:
        number=int(num)
        if number%2!=0:
                result.append(number)
print(result)
```

Output :

```
Enter integers separated by comma: 5,6,7,8,9,10
[5, 7, 9]
```

Result : The program is successfully executed and the output is verified.

Experiment No : 11

Date: 24/10/2024

Aim : Accept a list of words and return the length of the longest word.

Pseudocode :

```
DISPLAY "Enter the list of words seperated by commas: "
GET str
SPLIT str by ‘,’

SET list = empty list

FOR each word i in str DO
    SET list.append(i)
    SET longest_length=0
    FOR each list1 in list DO
        IF length of list1 is greater than longest_length THEN
        SET longest_length = length of list1

DISPLAY " The longest word is {list1} and the length is {longest_length}"
```

Method :

Functions	Description	Syntax
str.len()	used to determine the length of an object.	len(object)

Souce Code :

```
str=input("Enter the list of words seperated by commas: ")
words=str.split(',')
list=[]
for i in words:
        list.append(i)
longest_length=0
for list1 in list:
        if len(list1)>longest_length:
                long=list1
                longest_length=len(list1)
print(f"The longest word is {list1} and the length is {longest_length}")
```

Output :

```
Enter the list of words seperated by commas: apple,orange,pineapple,watermelon
The longest word is watermelon and the length is 10
```

Result : The program is successfully executed and the output is verified.

Experiment No : 12

Date: 24/10/2024

Aim : Write a program to prompt the user to enter two lists of integers and check
(a) Whether lists are of the same length.
(b) Whether the list sums to the same value.
(c) Whether any value occurs in both Lists

Pseudocode :

```
DISPLAY " Enter the no of integers for list1: "
GET n1
SET list1 = empty list

FOR i FROM 0 TO n1-1 DO
    DISPLAY "Enter the Integers: "
    GET num1
    APPEND num1 to list1

DISPLAY "Enter the number of Integers for List-2: "
GET n2
SET list2 = empty list

FOR i FROM 0 TO n2-1 DO
    DISPLAY "Enter the Integers: "
    GET num2
    APPEND num2 to list2

IF length of list1 is equal to length of list2 THEN
    DISPLAY "Length of Both Lists are Same"
ELSE
    DISPLAY "Length of Both Lists are Different"

IF sum of list1 is equal to sum of list2 THEN
    DISPLAY "Sum of both Lists are Same"
ELSE
    DISPLAY "Sum of both Lists are Different"

SET common_val = intersection of list1 and list2

IF common_val is not empty THEN
    DISPLAY common_val, "is the common value of both List"
ELSE
```

```
    DISPLAY "Common values does not exist "
```

Method :

Functions	Description	Syntax
sum()	used to calculate the total of all numeric values in an iterable (like a list, tuple, or set).	sum(iterable, start=0)

Souce Code :

```
n1=int(input("Enter the no of integers for list1: "))
list1=[]
for i in range(n1):
        num1=int(input("Enter the integers: "))
        list1.append(num1)
n2=int(input("Enter the no of integers for list2: "))
list2=[]
for i in range(n2):
        num2=int(input("Enter the integers: "))
        list2.append(num2)
if len(list1) == len(list2):
        print("Length of Both lists are same")
else:
          print("Length of Both lists are Different")
if sum(list1) == sum(list2):
        print("Sum of Both lists are same")
else:
          print("Sum of Both lists are Different")
common_val=set(list1).intersection(list2)
if common_val:
        print(f"{common_val} is the common value of Both lists")
else:
          print("Common values doesnot exit..")
```

Output :

```
Enter the no of integers for list1: 4
Enter the integers: 12
Enter the integers:14
Enter the integers:17
Enter the integers:22
Enter the no of integers for list2:3
Enter the integers:11
Enter the integers:17
Enter the integers:32
Length of Both lists are Different
Sum of Both lists are Different
{17} is the common value of Both lists
```

Result : The program is successfully executed and the output is verified.

Experiment No :13

Date: 24/10/2024

Aim :

Write a Python program to count the occurrences of each word in a line of text.
Sample input : the quick brown fox jumps over the lazy dog

Pseudocode :

```
DISPLAY "Enter the line of text: "
GET text
CONVERT text to lowercase

SPLIT text into words by spaces

SET word_count = empty dictionary

FOR each word in words DO
    IF word is in word_count THEN
        Increment word_count[word] by 1
    ELSE
        SET word_count[word] = 1

DISPLAY "Word occurence: ",word_count
```

Souce Code :

```
text=input("Enter a line of text: ")
words=text.split()
word_count={}
for i in words:
        words=i.lower()
        if i in word_count:
                word_count[words]+=1
        else:
                word_count[words]=1
print("Word occurence: ",word_count)
```

Output :

```
Enter a line of text: his name is ashin and his hobbies are football and efootball
```

```
Word occurrence: {‘his’: 2, ‘name’: 1, ‘is’: 1, ‘ashin’: 1, ‘and’: 2, ‘hobbies’: 1, ‘are’: 1,
‘football’: 1, ‘efootball’: 1}
```

Result : The program is successfully executed and the output is verified.

Experiment No : 14

Date: 24/10/2024

Aim :

List comprehensions:
(a) Generate positive list of numbers from a given list of integers
(b) Square of N numbers
(c) Form a list of vowels selected from a given word
(d) Form a list ordinal value of each element of a word (Hint: use ord() to get ordinal values)

Pseudocode :

```
SET numbers =[-10,15,-3,7,-25,18,0]

SET positive_num = num for num in numbers that are greater than 0

DISPLAY "Postive numbers are: ",positive_num

SET n=5
SET square=[num**2 for num in range(1,n+1)]

DISPLAY "Square of 1st 5 number:",square

SET words="comprehension"
SET vowels= char for char in words if char in 'aeiou'

SET wd = "hello"
SET vow = set comprehension to get vowels from wd that are in ['a', 'e', 'i', 'o', 'u']

DISPLAY "Vowels in the word is {vowels} is : ",vowels

SET word="hello"
SET ordinal_val = list comprehension to get the ordinal values of each character in wordd

DISPLAY "Ordinal values for each char in the word: ",ordinal_val
```

Method :

Functions	Description	Syntax
ord()	returns the Unicode code point (integer representation) of a given character.	ord(char)

Souce Code :

```
numbers=[-10,15,-3,7,-25,18,0]
positive_num=[num for num in numbers if num>0]
print("Postive numbers are: ",positive_num)
n=5
square=[num**2 for num in range(1,n+1)]
print("Square of 1st 5 number:",square)
words="comprehension"
vowels=[char for char in words if char in 'aeiou']
print(f"Vowels in the word is {vowels} is : ",vowels)
word="hello"
ordinal_val=[ord(char) for char in word]
print("Ordinal values for each char in the word: ",ordinal_val)
```

Output :

```
Positive numbers are: [15, 7, 18]
Square of 1st 5 number: [1, 4, 9, 16, 25]
Vowels in the word is comprehension is : [‘o’, ‘e’, ‘e’,   ‘i’, ‘o’]
Ordinal values for each char in the word: [104, 101, 108, 108, 111]
```

Result : The program is successfully executed and the output is verified.

Experiment No : 15

Date: 24/10/2024

Aim :

Sort dictionary in ascending and descending order.

Pseudocode :

```
SET mydic = {'apple':3,'orange':7,'kiwi':2,'banana':5}

SET ascending_keys = sorted keys of my_dict in ascending order
SET descending_keys = sorted keys of my_dict in descending order

DISPLAY "Ascending order of the keys:{ascending_keys}"
DISPLAY "Descending order of the keys:{descending_keys}"

SET ascending_values = sorted values of my_dict in ascending order
SET descending_values = sorted values of my_dict in descending order

DISPLAY " Ascending order of the values:{ascending_values}"
DISPLAY "Descending order of the values:{descending_values}"
```

Method :

Functions	Description	Syntax
key()	This method returns a view object displaying all the keys in the dictionary.	dict.key()
value()	This method returns a view object displaying all the values in the dictionary.	dict.value()
sorted()	returns a sorted list of the specified iterable's elements.	sorted(dict.keys)

Souce Code :

```
mydic={'apple':3,'orange':7,'kiwi':2,'banana':5}
ascending_keys=sorted(mydic.keys())
descending_keys=sorted(mydic.keys(),reverse=True)
print(f"Ascending order of the keys:{ascending_keys}")
print(f"Descending order of the keys:{descending_keys}")
ascending_values=sorted(mydic.values())
descending_values=sorted(mydic.values(),reverse=True)
```

```
print(f"Ascending order of the values:{ascending_values}")
print(f"Descending order of the values:{descending_values}")
```

Output :

```
Ascending order of the keys:[ 'apple','banana','kiwi','orange']
Descending order of the keys:[ 'orange','kiwi','banana','apple']
Ascending order of the values:[2, 3, 5, 7]
Descending order of the values:[7, 5, 3, 2]
```

Result : The Program is successfully executed and the Output is verified.

Experiment No : 16

Date: 24/10/2024

Aim :

Merge two dictionaries.

Pseudocode :

```
SET dict1 = {'apple':4,'orange':2}
SET dict2 ={'mango':5,'kiwi':1}

DISPLAY dict1
DISPLAY dict2

UPDATE dict1 with the contents of dict2

DISPLAY "Merged Dictionary=", dict1
```

Method :

Functions	Description	Syntax
update()	method adds the key-value pairs from one dictionary to another.	dict1.update(dict2)

Souce Code :

```
dic1={'apple':4,'orange':2}
dic2={'mango':5,'kiwi':1}
print(dic1)
print(dic2)
dic1.update(dic2)
print("Merged Dictionary=",dic1
```

Output :

```
{'apple':4,'orange':2}
{'mango':5,'kiwi':1}
Merged Dictionary: {'apple':4,'orange':2, 'mango':5,'kiwi':1}
```

Result : The Program is successfully executed and the Output is verified.