Lab Manual 5.1

#Looping through list  
*'''list1 = ["cherry", "orange", "kiwi", "mango"]  
list2 = ["apple", "banana", "cherry"]  
for x in list2:  
 print(x)  
 list1.append(x)  
 print(list1)  
  
  
for i in range(4):  
 print(list1[i])  
  
for i in range(4):  
 print(list1[i], end=' ')'''*# Program to find the sum of all numbers stored in a list # List of numbers  
'''numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]  
# variable to store the sum  
sum = 0  
pro = 0  
# iterate over the list  
for val in numbers:  
 sum = sum + val  
 pro = sum \* val  
  
print("The sum is", sum)  
print("The product is", pro)'''  
#Calculate the square of each number of list  
  
'''numbers = [1, 2, 3, 4, 5]  
# iterate over each element in list num  
for i in numbers:  
 # \*\* exponent operator  
 square = i \*\* 2  
 print("Square of:", i, "is:", square)'''  
  
#Calculate the average of list of numbers  
  
'''numbers = {10, 20, 30, 40, 50}  
  
# definite iteration  
# run loop 5 times because list contains 5 items  
sum = 0  
for i in numbers:  
 sum = sum + i  
list\_size = len(numbers)  
average = sum / list\_size  
print(average)'''  
  
  
  
  
#Loop Through a Tuple  
  
'''thistuple = ("apple", "banana", "cherry")  
for x in thistuple:  
 print(x)'''  
  
# Program to find the sum of all numbers stored in a tuple  
'''numbers = (6, 5, 3, 8, 4, 2, 5, 4, 11)  
# variable to store the sum  
sum = 0  
# iterate over the list  
for val in numbers:  
 sum = sum + val  
 print("The sum is", sum)'''  
#Loop throu sets  
  
'''thisset = {"apple", "banana", "cherry"}  
for x in thisset:  
 print(x)  
# Program to find the sum of all numbers stored in a set  
numbers = {6, 5, 3, 8, 4, 2, 5, 4, 11}  
# variable to store the sum  
sum = 0  
# iterate over the list  
for val in numbers:  
 sum = sum + val  
 print("The sum is", sum)'''  
#Loop Through a Dictionary  
  
  
'''thisdict = {  
 "brand": "Ford",  
 "model": "Mustang",  
 "year": 1964  
}  
for x in thisdict:  
 print(x)  
  
for x in thisdict.values():  
 print(x)  
for x in thisdict.keys():  
 print(x)  
for x, y in thisdict.items():  
 print(x, y)'''  
  
# Program to find the sum of all values and keys stored in a dictionary  
  
'''numbers = {1:6, 2:5, 3:3, 4:8, 5:4, 6:2, 7:5, 8:4, 9:11}  
# variable to store the sum  
sumv = 0  
sumk = 0  
  
product = 0  
# iterate over the list  
for x in numbers.values():  
 sumv = sumv + x  
 print("The sum of values", sumv)  
for y in numbers.keys():  
 sumk = sumk + y  
 print("The sum of keys", sumk)  
  
  
print(sumv)  
print(sumk)  
product = sumv\*sumk  
print(product)'''  
  
  
#Looping Through a String  
  
'''for x in "Hello CyberSecurity":  
 print(x, end=' ')'''  
  
#With the break statement we can stop the loop before it has looped through all the items:  
  
'''fruits = ["apple", "banana", "cherry", "kiwi", "mango"]  
for x in fruits:  
 print(x)  
 if x == "banana":  
 break'''  
  
#Exit the loop when x is "banana", but this time the break comes before the print:  
  
'''fruits = ["apple", "banana", "cherry", "kiwi", "mango"]  
for x in fruits:  
 if x == "cherry":  
 break  
 print(x)'''  
#Example: break the loop if number a number is greater than 15  
  
'''numbers = [1, 4, 7, 8, 15, 20, 35, 45, 55]  
for i in numbers:  
 if i > 15:  
 # break the loop  
 break  
 else:  
 print(i)'''  
#With the continue statement we can stop the current iteration of the loop, and continue with the next:  
  
'''fruits = ["apple", "kiwi", "cherry"]  
for x in fruits:  
 if x == "kiwi":  
 continue  
 print(x)'''  
  
#With the continue statement for more than two values  
'''fruits = ["apple", "kiwi", "cherry", "mango"]  
for x in fruits:  
 if x == "kiwi":  
 continue  
 if x == "cherry":  
 continue  
 print(x)'''  
#Count the total number of ‘m’ in a given string.  
  
'''name = "mariya mennen"  
count = 0  
for char in name:  
 if char != 'm':  
 continue  
 else:  
 count = count + 1  
  
print('Total number of m is:', count)'''  
  
  
#To loop through a set of code a specified number of times, we can use the range() function,  
  
'''for x in range(6):  
 print(x)  
  
print("all done")'''  
  
#The range() function defaults to 0 as a starting value,  
# however it is possible to specify the starting value by adding a parameter:  
  
'''for x in range(2, 6):  
 print(x)  
  
for x in range(2, 6, 2):  
 print(x)'''  
  
#The range() function defaults to increment the sequence by 1, however it is possible  
# to specify the increment value by adding a third parameter: range(2, 30, 3):  
  
'''for x in range(2, 30, 3):  
 print(x)'''  
  
  
#The else keyword in a for loop specifies a block of code to be executed when the loop is finished:  
  
'''for x in range(6):  
 print(x)  
else:  
 print("Finally finished!")'''  
  
#The else block will NOT be executed if the loop is stopped by a break statement.#  
  
'''for x in range(6):  
 if x == 3: break  
 print(x)  
else:  
 print("Finally finished!")'''  
#Print all even and odd numbers  
  
'''for i in range(1, 11):  
 if i % 2 == 0:  
 print('Even Number:', i)  
 else:  
 print('Odd Number:', i)'''  
  
  
#A nested loop is a loop inside a loop. The "inner loop" will be executed one time for each iteration of the "outer loop":  
  
'''adj = ["red", "big", "tasty"]  
fruits = ["apple", "banana", "cherry"]  
  
for x in adj:  
 for y in fruits:  
 print(x, y)'''  
#Programs to print number pattern  
'''rows = 6  
# if you want user to enter a number, uncomment the below line  
# rows = int(input('Enter the number of rows'))  
# outer loop  
for i in range(rows):  
 # nested loop  
 for j in range(i):  
 # display number  
 print(i, end=' ')  
 # new line after each row  
 print('')'''  
  
#In each row, every next number is incremented by 1.  
  
'''rows = 6  
for i in range(1, rows):  
 print(i)  
 for j in range(1, i + 1):  
 print(j, end=' ')  
 print('')'''  
#star pattern  
# number of rows  
'''rows = 5  
for i in range(0, rows):  
 # nested loop for each column  
 for j in range(0, i + 1):  
 # print star  
 print("\*\*\*", end=' ')  
 # new line after each row  
 print("\r")'''  
#for loops cannot be empty, but if you for some reason have a for loop  
# with no content, put in the pass statement to avoid getting an error.  
  
'''for x in [0, 1, 2]:  
 pass'''  
  
'''num = [1, 4, 5, 3, 7, 8]  
for i in num:  
 # calculate multiplication in future if required  
 pass'''  
  
#Backward Iteration using the reversed() function  
  
# Reversed numbers using reversed() function  
'''list1 = [10, 20, 30, 40]  
for num in reversed(list1):  
 print(num)'''  
#Reverse for loop using range()  
  
'''print("Reverse numbers using for loop")  
num = 5  
# start = 5  
# stop = -1  
# step = -1  
for num in (range(num, -1, -1)):  
 print(num)'''