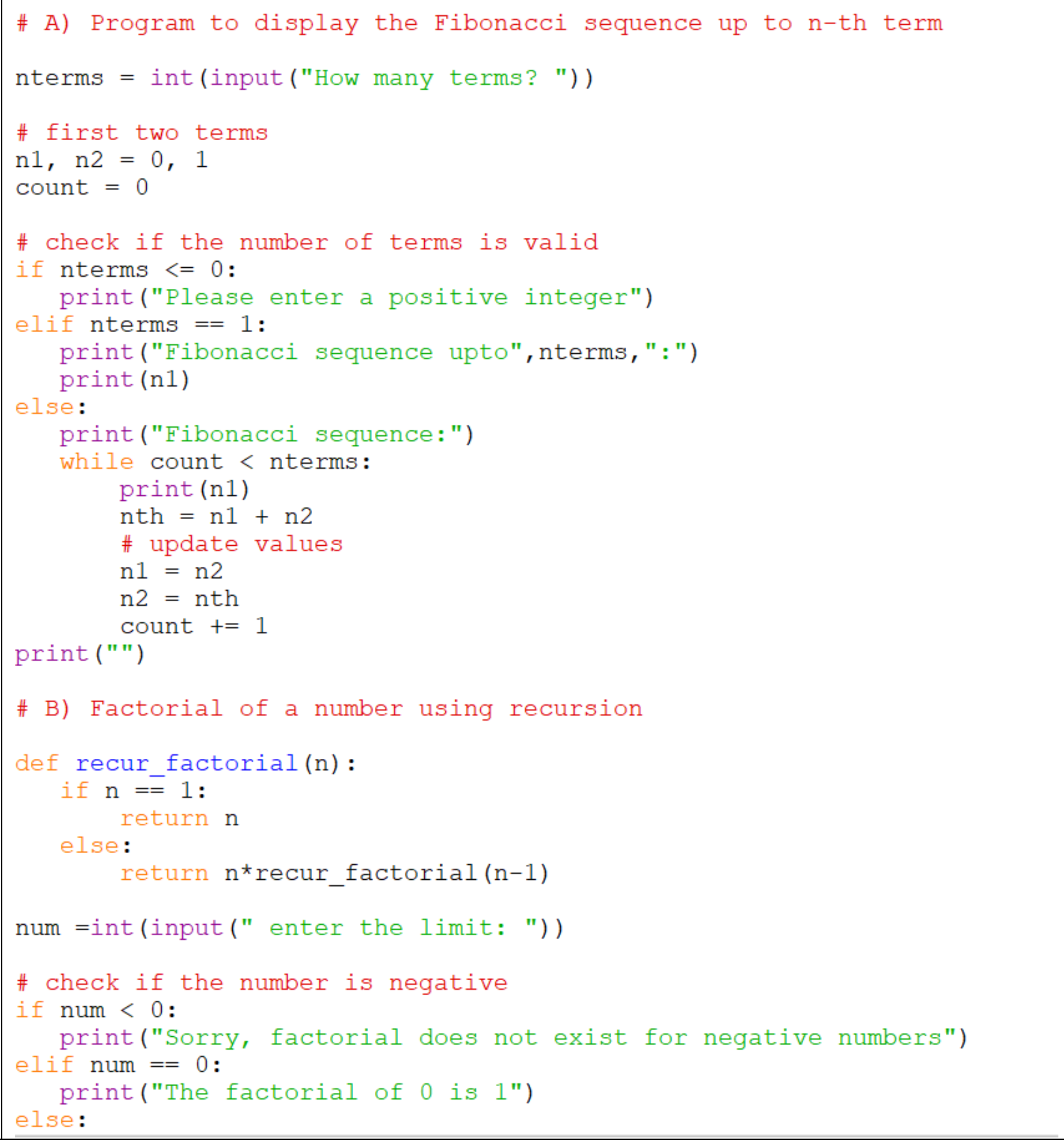
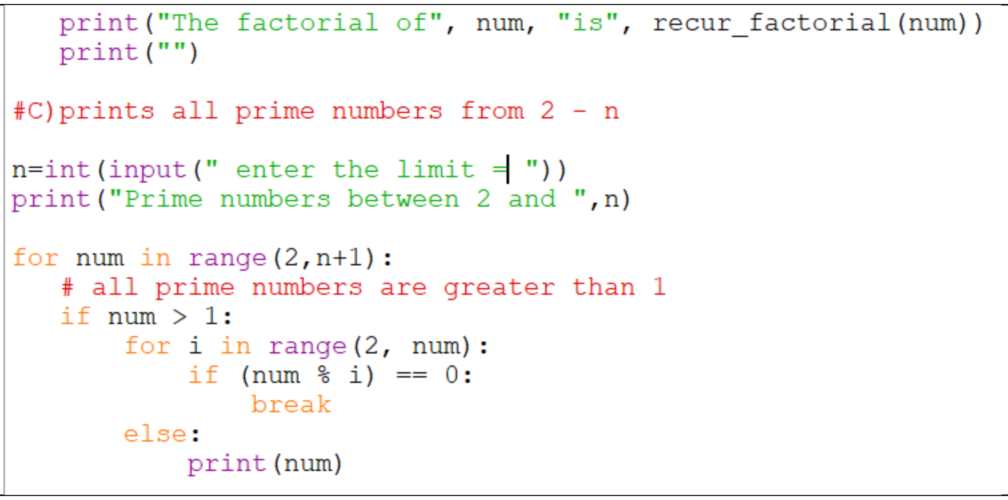
Week 4- Programs on Iterative constructs, Lists and Tuples

Program 1	a) Write a program to generate fibonacci series till n terms b) Find factorial of a number c) prints all prime numbers from 2 - n
	Algorithm: **a)** **1)first establish the values from which fibonacci sequences start** **2)then use identation and if and elif statements to code the fibonacci sequence** **b)** **1)state the values and print there is no factorial for negative numbers** **2)specify the condition for 0 and code the rest** **c)** **1)input the upper range** **2)calculate the range**
	Program with comments:

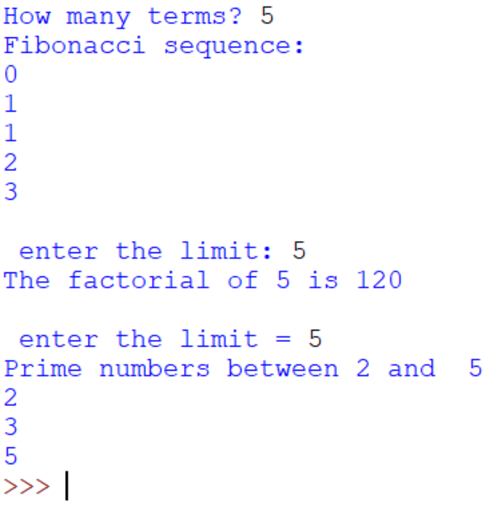


```
# A) Program to display the Fibonacci sequence up to n-th term

nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1)
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
print("")

# B) Factorial of a number using recursion

def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n*recur_factorial(n-1)

num =int(input(" enter the limit: "))

# check if the number is negative
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
```



```
   print("The factorial of", num, "is", recur_factorial(num))
   print("")

#C)prints all prime numbers from 2 - n

n=int(input(" enter the limit = "))
print("Prime numbers between 2 and ",n)

for num in range(2,n+1):
   # all prime numbers are greater than 1
   if num > 1:
       for i in range(2, num):
           if (num % i) == 0:
               break
       else:
           print(num)
```

Output:



```
How many terms? 5
Fibonacci sequence:
0
1
1
2
3

 enter the limit: 5
The factorial of 5 is 120

 enter the limit = 5
Prime numbers between 2 and  5
2
3
5
>>> |
```

Program 2

Write a python program to perform the following operations using given list as input:

a) **Given a heterogenous list, create separate lists for different types of data. Write a python program to achieve the same.**

b) **Sort in ascending and descending order**

i)list of strings ii) list of number

Algorithm:

a)

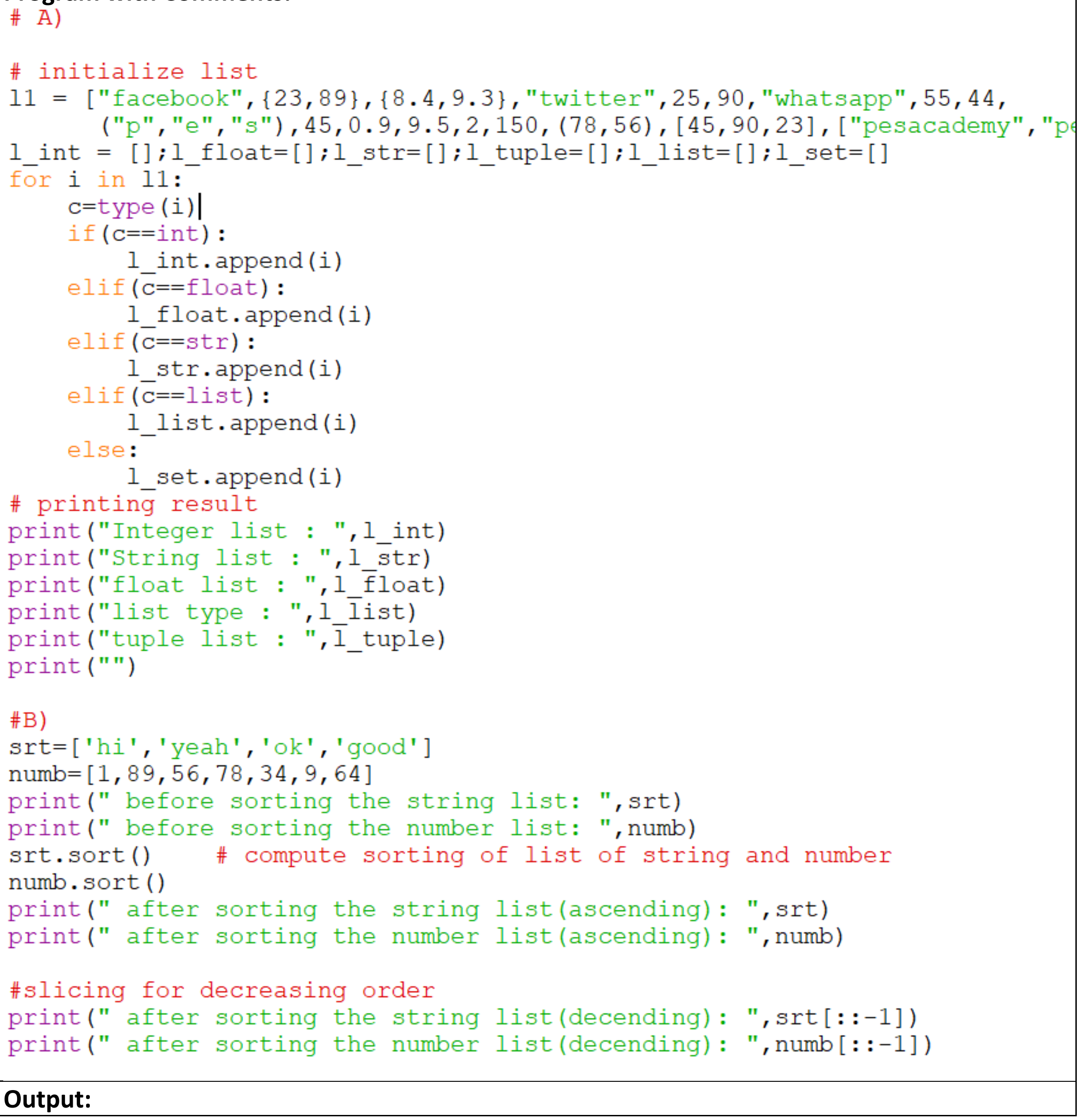
1)set up different lists for different data type

2)append suiting the data type for the values

3)print the values

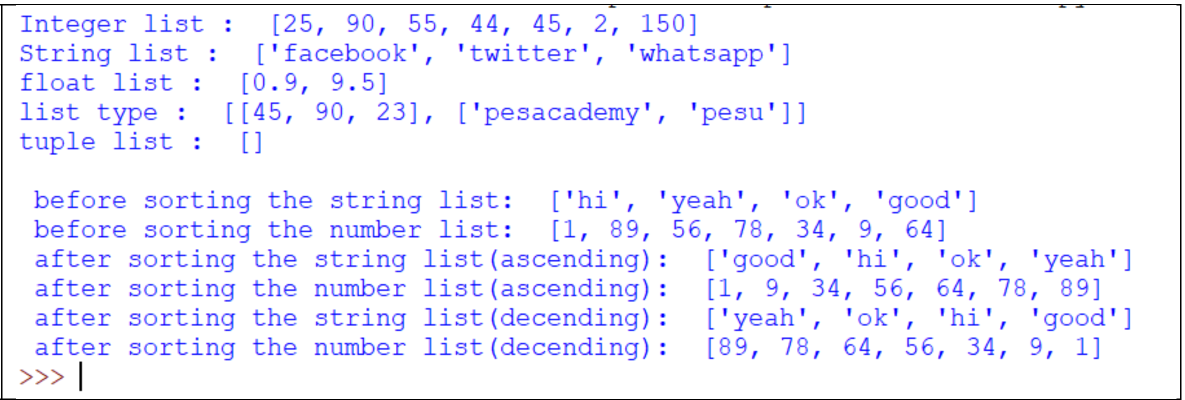
b)
1)list the names and numbers
2)use the sort function for sorting

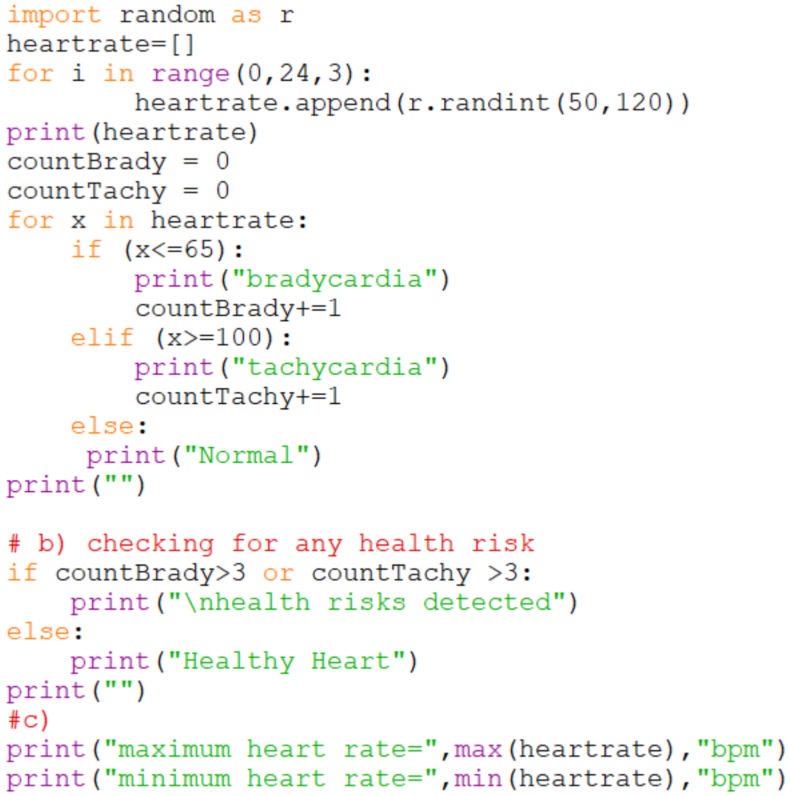
Program with Comments:



```
# A)

# initialize list
l1 = ["facebook",{23,89},{8.4,9.3},"twitter",25,90,"whatsapp",55,44,
      ("p","e","s"),45,0.9,9.5,2,150,(78,56),[45,90,23],["pesacademy","pe
l_int = [];l_float=[];l_str=[];l_tuple=[];l_list=[];l_set=[]
for i in l1:
    c=type(i)
    if(c==int):
        l_int.append(i)
    elif(c==float):
        l_float.append(i)
    elif(c==str):
        l_str.append(i)
    elif(c==list):
        l_list.append(i)
    else:
        l_set.append(i)
# printing result
print("Integer list : ",l_int)
print("String list : ",l_str)
print("float list : ",l_float)
print("list type : ",l_list)
print("tuple list : ",l_tuple)
print("")

#B)
srt=['hi','yeah','ok','good']
numb=[1,89,56,78,34,9,64]
print(" before sorting the string list: ",srt)
print(" before sorting the number list: ",numb)
srt.sort()    # compute sorting of list of string and number
numb.sort()
print(" after sorting the string list(ascending): ",srt)
print(" after sorting the number list(ascending): ",numb)

#slicing for decreasing order
print(" after sorting the string list(decending): ",srt[::-1])
print(" after sorting the number list(decending): ",numb[::-1])
```

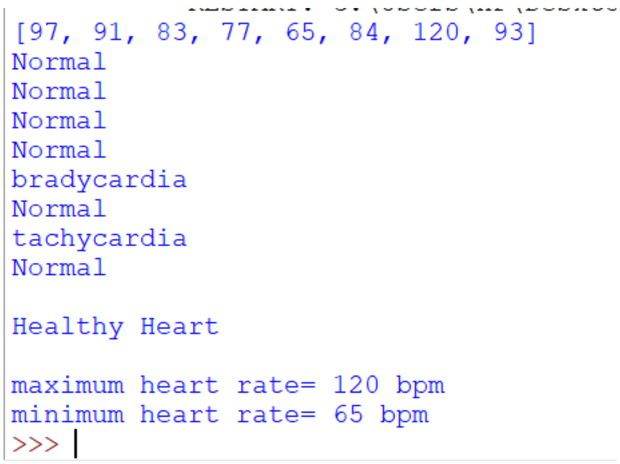
Output:

	 Integer list : [25, 90, 55, 44, 45, 2, 150] String list : ['facebook', 'twitter', 'whatsapp'] float list : [0.9, 9.5] list type : [[45, 90, 23], ['pesacademy', 'pesu']] tuple list : [] before sorting the string list: ['hi', 'yeah', 'ok', 'good'] before sorting the number list: [1, 89, 56, 78, 34, 9, 64] after sorting the string list(ascending): ['good', 'hi', 'ok', 'yeah'] after sorting the number list(ascending): [1, 9, 34, 56, 64, 78, 89] after sorting the string list(decending): ['yeah', 'ok', 'hi', 'good'] after sorting the number list(decending): [89, 78, 64, 56, 34, 9, 1] >>> |
Program 3	Generate heart rate randomly between 50 to 120 at time interval of 3 hours for 24 hours. **(i) If heart rate is between 50-65 print as bradycardia(lower heart rate) if greater than 100 print as tachycardia(higher heart rate). Else print as normal.** **(ii) Count number of Bradycardia and tachycardia if any of this is greater than 3 display as risk.** **(iii) Print the maximum heart rate and minimum heart rate**
	Algorithm: **I)import random to use random function** **2)use random values and storing** **3)calculate the maximum heart rate and minimum heart rate**
	Program with comments:



```
import random as r
heartrate=[]
for i in range(0,24,3):
        heartrate.append(r.randint(50,120))
print(heartrate)
countBrady = 0
countTachy = 0
for x in heartrate:
    if (x<=65):
        print("bradycardia")
        countBrady+=1
    elif (x>=100):
        print("tachycardia")
        countTachy+=1
    else:
     print("Normal")
print("")

# b) checking for any health risk
if countBrady>3 or countTachy >3:
    print("\nhealth risks detected")
else:
    print("Healthy Heart")
print("")
#c)
print("maximum heart rate=",max(heartrate),"bpm")
print("minimum heart rate=",min(heartrate),"bpm")
```

Output:



```
[97, 91, 83, 77, 65, 84, 120, 93]
Normal
Normal
Normal
Normal
bradycardia
Normal
tachycardia
Normal

Healthy Heart

maximum heart rate= 120 bpm
minimum heart rate= 65 bpm
>>> |
```

Program 4

Enter marks of students till you need to stop.

a) **Find maximum marks**

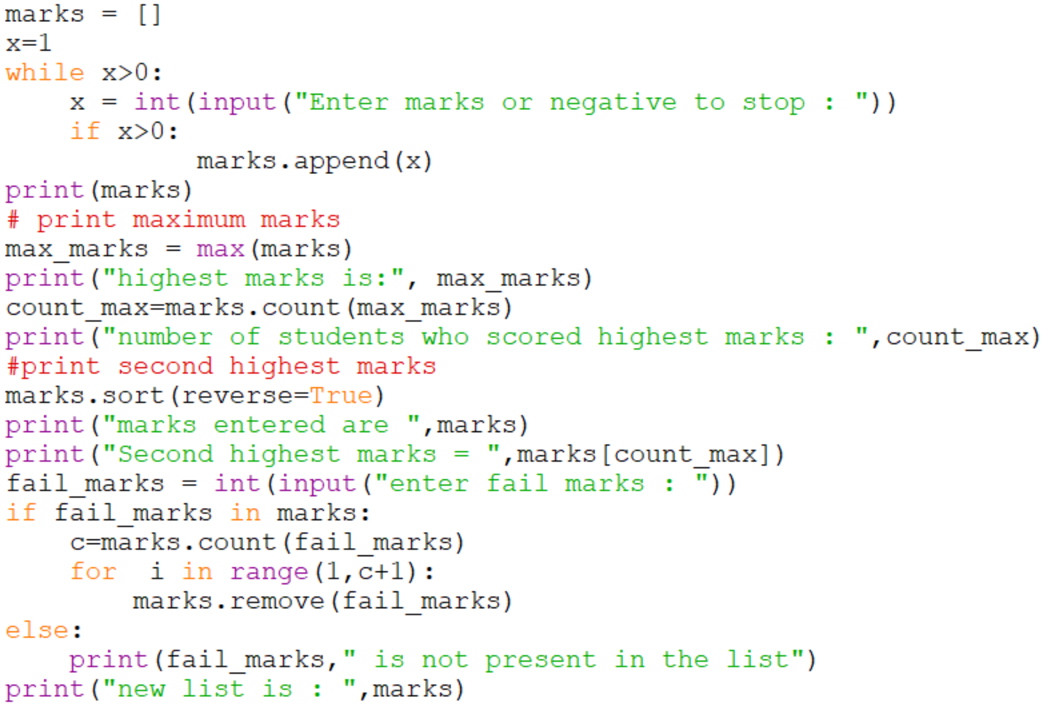
b) **Find number of students who have scored highest**

c) **Find second highest marks**

d) **Enter fail marks and remove if fail marks present in list**

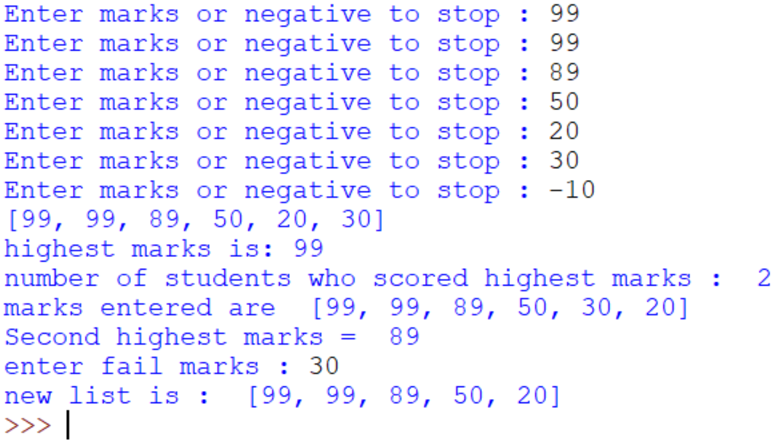
Algorithm:
1)input the marks
2)append the marks
3)use the sort function
4)delete the fail marks

Program with comment:



```
marks = []
x=1
while x>0:
    x = int(input("Enter marks or negative to stop : "))
    if x>0:
            marks.append(x)
print(marks)
# print maximum marks
max_marks = max(marks)
print("highest marks is:", max_marks)
count_max=marks.count(max_marks)
print("number of students who scored highest marks : ",count_max)
#print second highest marks
marks.sort(reverse=True)
print("marks entered are ",marks)
print("Second highest marks = ",marks[count_max])
fail_marks = int(input("enter fail marks : "))
if fail_marks in marks:
    c=marks.count(fail_marks)
    for  i in range(1,c+1):
        marks.remove(fail_marks)
else:
    print(fail_marks," is not present in the list")
print("new list is : ",marks)
```

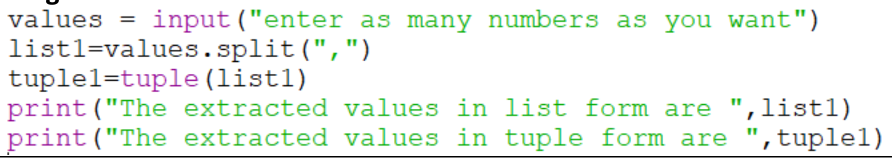
Output:

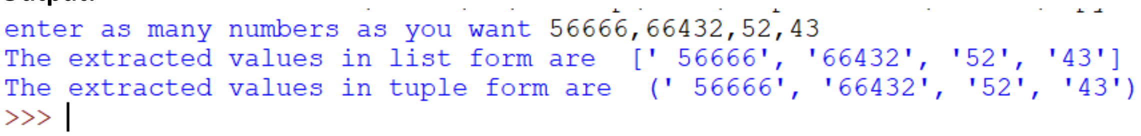


```
Enter marks or negative to stop : 99
Enter marks or negative to stop : 99
Enter marks or negative to stop : 89
Enter marks or negative to stop : 50
Enter marks or negative to stop : 20
Enter marks or negative to stop : 30
Enter marks or negative to stop : -10
[99, 99, 89, 50, 20, 30]
highest marks is: 99
number of students who scored highest marks :  2
marks entered are  [99, 99, 89, 50, 30, 20]
Second highest marks =  89
enter fail marks : 30
new list is :  [99, 99, 89, 50, 20]
>>> |
```

Program 5	Write a python program which accepts a sequence of comma-separated values from console

	and generate as a list and as a tuple.
	Algorithm: **1)input all the values** **2)use the split function** **3)print the extracted values**

	Program with comment
	
	`values = input("enter as many numbers as you want")`
	`list1=values.split(",")`
	`tuple1=tuple(list1)`
	`print("The extracted values in list form are ",list1)`
	`print("The extracted values in tuple form are ",tuple1)`

	Output:
	
	`enter as many numbers as you want 56666,66432,52,43`
	`The extracted values in list form are  [' 56666', '66432', '52', '43']`
	`The extracted values in tuple form are  (' 56666', '66432', '52', '43')`
	`>>> |`