

NAME: ARUNIMA SINGH THAKUR
REGISTRTION NO: 180905218
LAB NO: 2
LAB: DS
ROLL NO: 31
BATCH: C1

QUESTION 1:

Write a program to find the area of rectangle. Take input from user.

```
w = float(input('Please Enter the Width of a Rectangle: '))  
h = float(input('Please Enter the Height of a Rectangle: '))
```

```
Area = w * h
```

```
print("\n Area of a Rectangle is: %.2f" %Area)
```

A screenshot of a terminal window with a dark background. The title bar at the top reads "ugcse@prg28: ~/Desktop/180905218/lab2". Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal shows the command "python q1.py" being executed. The program prompts for the width and height of a rectangle. The user enters "2" for the width and "3" for the height. The program then outputs "Area of a Rectangle is: 6.00".

```
ugcse@prg28: ~/Desktop/180905218/lab2  
File Edit View Search Terminal Help  
ugcse@prg28:~/Desktop/180905218/lab2$ python q1.py  
Please Enter the Width of a Rectangle: 2  
Please Enter the Height of a Rectangle: 3  
  
Area of a Rectangle is: 6.00
```

QUESTION 2:

Write a program to swap the values of two variables.

```
x = int(input('Please Enter value 1: '))  
y = int(input('Please Enter value 2: '))
```

```
temp = x  
x = y  
y = temp
```

```
print("The value of x after swapping: {}".format(x))  
print("The value of y after swapping: {}".format(y))
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q2.py
Please Enter value 1: 3
Please Enter value 2: 4
The value of x after swapping: 4
The value of y after swapping: 3
```

QUESTION 3:

Write a program to find whether a number is even or odd.

```
num = int(input("Enter a number: "))
if (num % 2) == 0:
    print("{0} is Even".format(num))
else:
    print("{0} is Odd".format(num))
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q3.py
Enter a number: 4
4 is Even
```

QUESTION 4:

Write a program to check the largest among the given three numbers.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
    largest = num1
elif (num2 >= num1) and (num2 >= num3):
    largest = num2
else:
    largest = num3

print("The largest number is", largest)
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q4.py
Enter first number: 5
Enter second number: 3
Enter third number: 9
('The largest number is', 9.0)
```

QUESTION 5:

Write a program to demonstrate while loop with else.

```
n = 5
while n > 0:
    n = n - 1
    if n == 2:
        break
    print(n)
else:
    print("Loop is finished")
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q5.py
4
3
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 6:

Write a program to print the prime numbers for a user provided range.

```
upper = int(input("Enter upper range: "))
lower = int(input("Enter lower range: "))

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 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)
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q6.py
Enter upper range: 67
Enter lower range: 23
('Prime numbers between', 23, 'and', 67, 'are:')
23
29
31
37
41
43
47
53
59
61
67
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 7:

Write a program to demonstrate List functions and operations.

```
arr = []
print(f"List: {arr}\n\nAppend 1,1,2,3: ")
arr.append(1)
arr.append(1)
arr.append(3)
arr.append(2)
print(f"{arr}\n\nCount of 1: {arr.count(1)}")
print(f"\nIndex of 2: {arr.index(2)}")
arr.reverse()
print(f"\nReverse list: {arr}")
arr.sort()
print(f"\nSort list: {arr}")
print(f"\nClear list")
arr.clear()
print(f"{arr}")
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 q7.py
List: []

Append 1,1,2,3:
[1, 1, 3, 2]

Count of 1: 2

Index of 2: 3

Reverse list: [2, 3, 1, 1]

Sort list: [1, 1, 2, 3]

Clear list
[]
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 8:

Consider the tuple(1,3,5,7,9,2,4,6,8,10). Write a program to print half its values in one line and the other half in the next line.

```
tp = (1,3,5,7,9,2,4,6,8,10)
tp1 = tp[:5]
tp2 = tp[5:]
print tp1
print tp2
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q8.py
(1, 3, 5, 7, 9)
(2, 4, 6, 8, 10)
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 9:

Consider the tuple (12, 7, 38, 56, 78). Write a program to print another tuple whose values are even number in the given tuple.

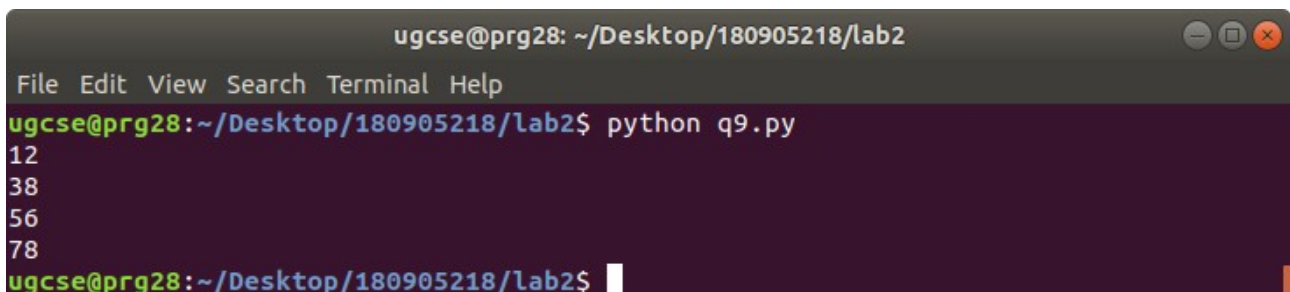
```
thistuple=(12, 7, 38, 56, 78 )
t=list(thistuple)
x=[]
i=0
```

```

while i<len(thistuple):
    if thistuple[i]%2==0:
        x.append(thistuple[i])
    i+=1

y=tuple(x)
i=0
while i<len(y):
    print y[i]
    i+=1

```



```

ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q9.py
12
38
56
78
ugcse@prg28:~/Desktop/180905218/lab2$

```

QUESTION 10:

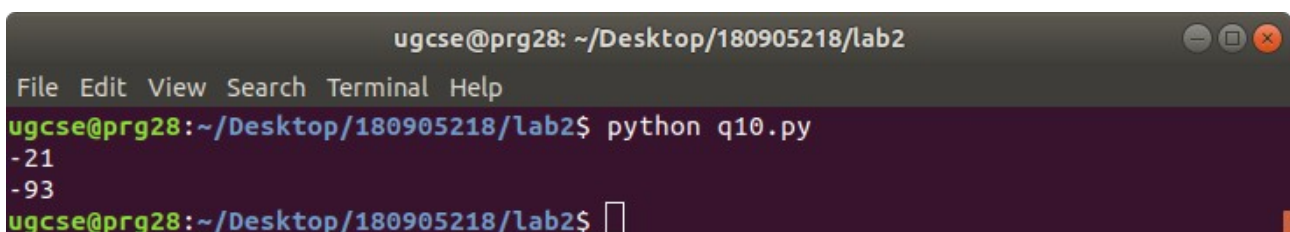
Write a Python program to print negative Numbers in a List using for loop. Eg. [11, -21, 0, 45, 66, -93].

```
x=[11, -21, 0, 45, 66, -93]
```

```

for i in x:
    if i<0:
        print i

```



```

ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q10.py
-21
-93
ugcse@prg28:~/Desktop/180905218/lab2$

```

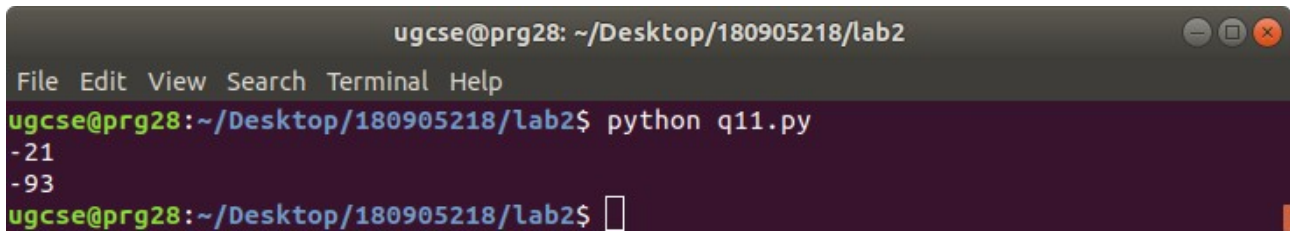
QUESTION 11:

Write a program to print negative Numbers in a List using while loop.

```
x=[11, -21, 0, 45, 66, -93]
```

```
i=0
```

```
while i<len(x):
    if x[i]<0:
        print x[i]
    i+=1
```



```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q11.py
-21
-93
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 12:

Write a Python program to count positive and negative numbers in a List.

```
list1 = [10, -21, 4, -45, 66, -93, 1]
```

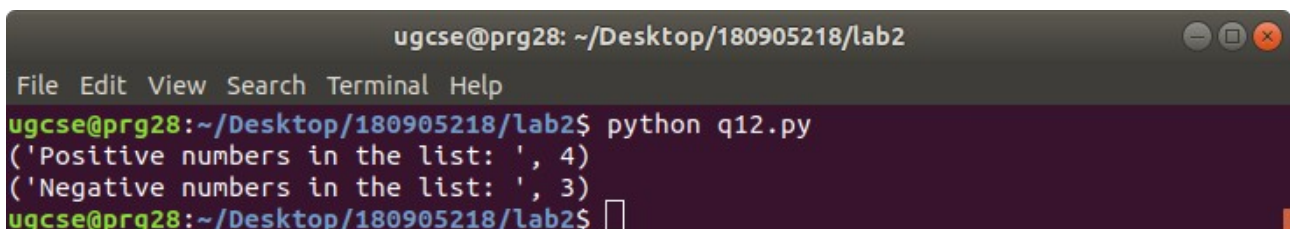
```
pos_count, neg_count = 0, 0
```

```
for num in list1:
```

```
    # checking condition
    if num >= 0:
        pos_count += 1
```

```
    else:
        neg_count += 1
```

```
print("Positive numbers in the list: ", pos_count)
print("Negative numbers in the list: ", neg_count)
```



```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q12.py
('Positive numbers in the list: ', 4)
('Negative numbers in the list: ', 3)
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 13:

Write a Python program to remove all even elements from a list .

```
list = [11, 22, 33, 44, 55]
```



```
print "Original list:"
print list

for i in list:
    if(i%2 == 0):
        list.remove(i)

print "list after removing EVEN numbers:"
print list
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q13.py
Original list:
[11, 22, 33, 44, 55]
list after removing EVEN numbers:
[11, 33, 55]
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 14:

Define a dictionary containing Students data {Name, Height, Qualification}.

- Convert the dictionary into DataFrame
- Declare a list that is to be converted into a new column (Address)
- Using 'Address' as the column name and equate it to the list and display the result.

```
import pandas as pd
data = [['Dinesh',178,"B.Tech"],['Nithya',128,"M.Tech"],['Raji',135,"B.Tech"]]
df = pd.DataFrame(data,columns=['Name','Height','Qualification'])
address=["Ranchi","Delhi","Mumbai"]
df["Address"]=address
print df
```



```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q14.py
   Name  Height Qualification Address
0  Dinesh    178        B.Tech  Ranchi
1  Nithya    128        M.Tech   Delhi
2    Raji    135        B.Tech  Mumbai
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 15:

Define a dictionary containing Students data {Name, Height, Qualification}.

a) Convert the dictionary into DataFrame

b) Use DataFrame.insert() to add a column and display the result.

```
import pandas as pd
```

```
data = [['Dinesh',178,"B.Tech"],['Nithya',128,"M.Tech"],['Raji',135,"B.Tech"]]
```

```
df = pd.DataFrame(data,columns=['Name','Height','Qualification'])
```

```
df.insert(3,"roll",[3,5,6])
```

```
print df
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python q15.py
   Name  Height Qualification  roll
0  Dinesh    178        B.Tech     3
1  Nithya    128        M.Tech     5
2    Raji    135        B.Tech     6
ugcse@prg28:~/Desktop/180905218/lab2$
```

SAMPLE QUESTIONS:

QUESTION 1:

#Panda Series with sum,max and min functions

```
import pandas as pd
```

```
import numpy as np
```

```
s=pd.Series([3,9,-2,10,5])
```

```
print("Total Sum: ",s.sum())
```

```
print("Minimum: ",s.min())
```

```
print("Maximum: ",s.max())
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 Sq1.py
Total Sum: 25
Minimum: -2
Maximum: 10
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 2:

#Creating a Data Frame

```
import pandas as pd
```

```
data=[['Dinesh',10],['Nithya',12],['Raji',13]]
```

```
df=pd.DataFrame(data,columns=['Name','Age'])
```

```
print("Normal Dataframe")
```

```
print(df)
```

```
print("\nIndexed Data Frame")
```

```
data = {'Name':['Kavitha', 'Sudha', 'Raju','Vignesh'],'Age':[28,34,29,42]}
```

```
df = pd.DataFrame(data, index=['rank-1','rank-2','rank-3','rank-4'])
```

```
print(df)
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 Sq2.py
Normal Dataframe
   Name  Age
0  Dinesh   10
1  Nithya   12
2    Raji   13

Indexed Data Frame
   rank-1  Age  Name
rank-1    28  Kavitha
rank-2    34   Sudha
rank-3    29    Raju
rank-4    42  Vignesh
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 3:

#Create Dataframe using Dictionary

```
import pandas as pd
```

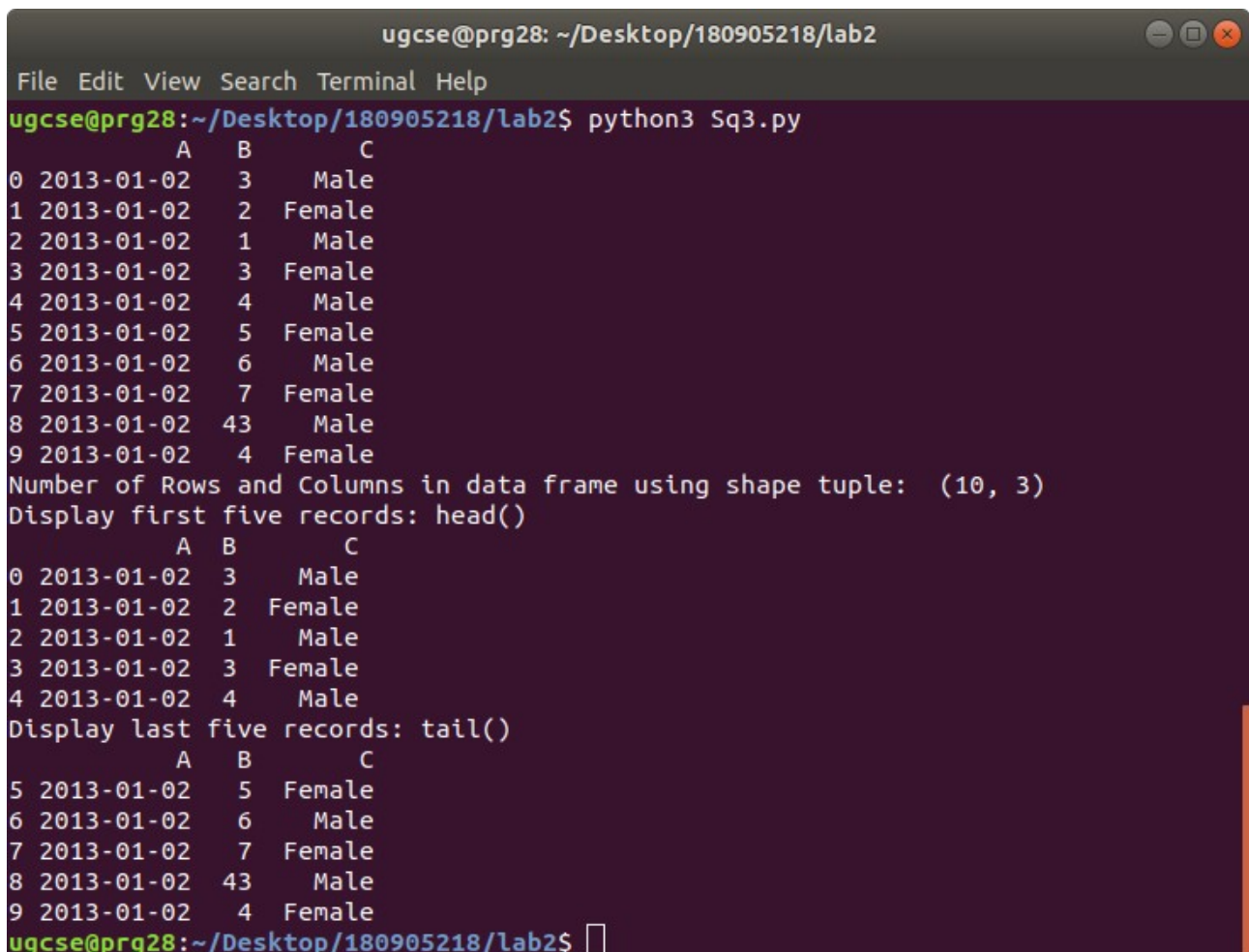
```
import numpy as np
```

```
df1=pd.DataFrame({
    'A':pd.Timestamp('20130102'),
    'B':np.array([3,2,1,3,4,5,6,7,43,4]),
```

```

'C':pd.Categorical(['Male','Female','Male','Female','Male','Female','Male','Female','Male','Female'])
})
print(df1)
print("Number of Rows and Columns in data frame using shape tuple: ",df1.shape)
print("Display first five records: head()")
print(df1.head())
print("Display last five records: tail()")
print(df1.tail())

```



```

ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 Sq3.py
      A  B  C
0 2013-01-02  3  Male
1 2013-01-02  2  Female
2 2013-01-02  1  Male
3 2013-01-02  3  Female
4 2013-01-02  4  Male
5 2013-01-02  5  Female
6 2013-01-02  6  Male
7 2013-01-02  7  Female
8 2013-01-02 43  Male
9 2013-01-02  4  Female
Number of Rows and Columns in data frame using shape tuple: (10, 3)
Display first five records: head()
      A  B  C
0 2013-01-02  3  Male
1 2013-01-02  2  Female
2 2013-01-02  1  Male
3 2013-01-02  3  Female
4 2013-01-02  4  Male
Display last five records: tail()
      A  B  C
5 2013-01-02  5  Female
6 2013-01-02  6  Male
7 2013-01-02  7  Female
8 2013-01-02 43  Male
9 2013-01-02  4  Female
ugcse@prg28:~/Desktop/180905218/lab2$ 

```

QUESTION 4:

```

#Creating Dataframe using randomn values
import pandas as pd
import numpy as np
#generates 100 randomn days
dates=pd.date_range('20130101',periods=100)
df=pd.DataFrame(np.random.randn(100,4),index=dates,columns=list('ABCD'))
print(df)

```

```
print("First five records of data frame: head()")
print(df.head())
print("Last five records of data frame: tail()")
print(df.tail())
print("To print the list of index: index tuple object")
print(df.index)
print("To view column names: columns tuple")
print(df.columns)
print("Sorting by Axis: ")
print(df.sort_index(axis=1,ascending=False))
print("Slicing the rows: Displaying first 3 rows")
print(df[0:3])
print("Slicing the rows with index range: 2013-01-05 TO 2013-01-10")
print(df['20130105':'20130110'])
print("Slicing with row and Column Index: df.iloc[0]- fetches first row")
print("Third row with df.iloc[2]\n",df.iloc[2])
print('Selecting a single column of A,B,C D: "A"\n',df['A'])
print('Selecting more than one column of A,B,C D: "A" & "B"\n',df[['A','B']])
print('Selecting more than one column of A,B,C D: "A" & "B" & "C"\n',df[['A','B','C']])
print("Selecting 2 or more columns with fixed number of rows\n",df[['A','B','C']][:5])
```

ugcse@prg28:~/Desktop/180905218/lab2\$ python3 Sq4.py

	A	B	C	D
2013-01-01	-0.019751	-2.054570	0.478385	-0.268219
2013-01-02	0.186694	-0.021699	0.115422	-0.959345
2013-01-03	1.087598	-2.046070	-0.644288	-0.017466
2013-01-04	1.078766	0.122688	1.316727	-1.906883
2013-01-05	0.007092	0.224715	-0.614482	-0.389969
2013-01-06	1.353015	-0.727713	0.395711	0.322069
2013-01-07	0.574828	0.679433	0.051841	-1.111272
2013-01-08	0.038109	0.508631	0.090748	1.842950
2013-01-09	-0.778788	-1.859258	-0.046374	-1.677985
2013-01-10	0.799996	0.215948	0.415404	0.441732
2013-01-11	-0.537291	-0.706203	-2.121600	1.078595
2013-01-12	-0.887146	1.101918	0.099289	0.875734
2013-01-13	0.923141	-0.983542	-0.031506	-1.058807
2013-01-14	-0.669893	-0.710936	0.860274	-1.906660
2013-01-15	-0.842807	-0.363456	0.238782	1.397176
2013-01-16	0.950945	-1.012210	0.051244	1.061093
2013-01-17	0.185849	-0.369138	1.358890	1.969528
2013-01-18	2.493928	-1.260515	-0.173396	-0.435309
2013-01-19	0.278692	0.538613	0.135534	-0.100728
2013-01-20	1.290883	0.059049	-0.608971	-1.993406
2013-01-21	-0.403839	-0.341072	-0.506873	1.780078
2013-01-22	-0.228340	-0.775293	0.446363	0.977908
2013-01-23	-0.876325	0.713774	1.482578	1.487934
2013-01-24	0.508370	-0.623169	-0.906386	-0.273335
2013-01-25	0.347697	-0.946678	0.843442	-0.151413
2013-01-26	-1.090363	-1.239445	-0.569390	0.606900
2013-01-27	0.330988	0.237725	-1.860511	1.095147
2013-01-28	-1.081083	1.708078	-1.694962	-1.543907
2013-01-29	0.333810	-0.785296	-0.257287	0.322648
2013-01-30	-0.033787	-0.493696	0.539905	-1.031483
...
2013-03-12	-1.301861	-0.227096	-1.226071	0.298662
2013-03-13	-1.374735	1.723079	-0.399122	0.104361
2013-03-14	-1.898323	0.617291	1.442930	0.649129
2013-03-15	0.160007	0.239464	-0.847818	0.948464
2013-03-16	0.046592	0.057098	0.058319	1.082121
2013-03-17	0.202484	-2.108420	-0.756871	-0.387863
2013-03-18	-1.130446	1.034407	1.236570	-1.288389
2013-03-19	0.247633	-0.997183	0.005466	-0.186742
2013-03-20	0.509432	-0.141482	0.319239	-0.428923
2013-03-21	0.476004	-0.438757	0.544433	0.435153
2013-03-22	-0.145042	-1.710034	-0.050870	-0.489809


```

'2013-02-10', '2013-02-11', '2013-02-12', '2013-02-13',
'2013-02-14', '2013-02-15', '2013-02-16', '2013-02-17',
'2013-02-18', '2013-02-19', '2013-02-20', '2013-02-21',
'2013-02-22', '2013-02-23', '2013-02-24', '2013-02-25',
'2013-02-26', '2013-02-27', '2013-02-28', '2013-03-01',
'2013-03-02', '2013-03-03', '2013-03-04', '2013-03-05',
'2013-03-06', '2013-03-07', '2013-03-08', '2013-03-09',
'2013-03-10', '2013-03-11', '2013-03-12', '2013-03-13',
'2013-03-14', '2013-03-15', '2013-03-16', '2013-03-17',
'2013-03-18', '2013-03-19', '2013-03-20', '2013-03-21',
'2013-03-22', '2013-03-23', '2013-03-24', '2013-03-25',
'2013-03-26', '2013-03-27', '2013-03-28', '2013-03-29',
'2013-03-30', '2013-03-31', '2013-04-01', '2013-04-02',
'2013-04-03', '2013-04-04', '2013-04-05', '2013-04-06',
'2013-04-07', '2013-04-08', '2013-04-09', '2013-04-10'],
dtype='datetime64[ns]', freq='D')

```

To view column names: columns tuple

Index(['A', 'B', 'C', 'D'], dtype='object')

Sorting by Axis:

	D	C	B	A
2013-01-01	-0.268219	0.478385	-2.054570	-0.019751
2013-01-02	-0.959345	0.115422	-0.021699	0.186694
2013-01-03	-0.017466	-0.644288	-2.046070	1.087598
2013-01-04	-1.906883	1.316727	0.122688	1.078766
2013-01-05	-0.389969	-0.614482	0.224715	0.007092
2013-01-06	0.322069	0.395711	-0.727713	1.353015
2013-01-07	-1.111272	0.051841	0.679433	0.574828
2013-01-08	1.842950	0.090748	0.508631	0.038109
2013-01-09	-1.677985	-0.046374	-1.859258	-0.778788
2013-01-10	0.441732	0.415404	0.215948	0.799996
2013-01-11	1.078595	-2.121600	-0.706203	-0.537291
2013-01-12	0.875734	0.099289	1.101918	-0.887146
2013-01-13	-1.058807	-0.031506	-0.983542	0.923141
2013-01-14	-1.906660	0.860274	-0.710936	-0.669893
2013-01-15	1.397176	0.238782	-0.363456	-0.842807
2013-01-16	1.061093	0.051244	-1.012210	0.950945
2013-01-17	1.969528	1.358890	-0.369138	0.185849
2013-01-18	-0.435309	-0.173396	-1.260515	2.493928
2013-01-19	-0.100728	0.135534	0.538613	0.278692
2013-01-20	-1.993406	-0.608971	0.059049	1.290883
2013-01-21	1.780078	-0.506873	-0.341072	-0.403839
2013-01-22	0.977908	0.446363	-0.775293	-0.228340
2013-01-23	1.487934	1.482578	0.713774	-0.876325
2013-01-24	-0.273335	-0.906386	-0.623169	0.508370

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
2013-01-28 -1.081083 1.708078 -1.694962
2013-01-29 0.333810 -0.785296 -0.257287
2013-01-30 -0.033787 -0.493696 0.539905
...
2013-03-12 -1.301861 -0.227096 -1.226071
2013-03-13 -1.374735 1.723079 -0.399122
2013-03-14 -1.898323 0.617291 1.442930
2013-03-15 0.160007 0.239464 -0.847818
2013-03-16 0.046592 0.057098 0.058319
2013-03-17 0.202484 -2.108420 -0.756871
2013-03-18 -1.130446 1.034407 1.236570
2013-03-19 0.247633 -0.997183 0.005466
2013-03-20 0.509432 -0.141482 0.319239
2013-03-21 0.476004 -0.438757 0.544433
2013-03-22 -0.145042 -1.710034 -0.050870
2013-03-23 0.575750 -1.487796 0.880937
2013-03-24 0.124848 -2.158743 1.194387
2013-03-25 -0.445269 -0.289772 0.691170
2013-03-26 -2.039138 0.547323 0.577954
2013-03-27 -0.363702 -0.912117 -1.178699
2013-03-28 -0.613957 0.357009 1.149644
2013-03-29 0.232121 -0.575605 -0.558997
2013-03-30 0.180859 -1.645830 0.124795
2013-03-31 -0.498980 -0.943599 -0.304986
2013-04-01 0.575341 -0.630229 0.381567
2013-04-02 2.853141 2.246705 -1.333071
2013-04-03 0.205967 1.951057 -0.972441
2013-04-04 2.100618 1.239347 -0.446919
2013-04-05 -0.305214 0.201815 -0.264038
2013-04-06 -0.333357 1.117077 -0.040288
2013-04-07 0.002433 2.528941 0.998408
2013-04-08 -0.100424 -0.920347 -0.698820
2013-04-09 0.194823 2.340089 0.773445
2013-04-10 -0.253017 1.683650 0.223996

[100 rows x 3 columns]
Selecting 2 or more columns with fixed number of rows
      A      B      C
2013-01-01 -0.019751 -2.054570 0.478385
2013-01-02 0.186694 -0.021699 0.115422
2013-01-03 1.087598 -2.046070 -0.644288
2013-01-04 1.078766 0.122688 1.316727
2013-01-05 0.007092 0.224715 -0.614482
ugcse@prg28:~/Desktop/180905218/lab2$ QUESTION 2:
```

QUESTION 5:

#Boolean Indexing of Dataframes

```
import pandas as pd
```

```
import numpy as np
```

```
dates=pd.date_range('20210301',periods=10)
```

```
df=pd.DataFrame(np.random.randn(10,5),index=dates,columns=list('ABCDE'))
```

```
print("All Records with Value in column A as positive:\n",df[df.A>0])
```

```
print("Adding a sixth column:")
```



```

df['F']=['Male','Female','Male','Female','Male','Female','Male','Female','Male','Female']
print(df)
print("Replacing all values in a given column:\n")
df.loc[:, 'D']=np.array([5]*len(df))
print("Replaced COLUMN 'D' with all 5\n")
print(df)
print("sorting the values by Column B:\n",df.sort_values(by='B'))

```

```

ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 Sq5.py
All Records with Value in column A as positive:

```

	A	B	C	D	E
2021-03-01	0.073687	0.289288	-1.256938	0.376190	-1.159301
2021-03-03	1.299929	1.005249	0.985375	0.922945	1.654088
2021-03-09	1.375796	0.225218	-0.277615	-0.288386	-0.895532

```

Adding a sixth column:

```

	A	B	C	D	E	F
2021-03-01	0.073687	0.289288	-1.256938	0.376190	-1.159301	Male
2021-03-02	-0.508095	1.169204	-0.596631	0.132768	-0.227631	Female
2021-03-03	1.299929	1.005249	0.985375	0.922945	1.654088	Male
2021-03-04	-1.024657	-0.858542	-0.104141	0.335595	-0.927653	Female
2021-03-05	-1.550090	-1.572869	-1.488436	0.346251	-2.471808	Male
2021-03-06	-0.161631	-0.498060	-0.864079	0.687370	0.325774	Female
2021-03-07	-0.932733	1.564995	0.494762	-2.660015	-0.147785	Male
2021-03-08	-0.721355	0.452387	-0.557525	0.747480	-1.210777	Female
2021-03-09	1.375796	0.225218	-0.277615	-0.288386	-0.895532	Male
2021-03-10	-0.153365	0.558645	0.022999	-0.468793	-0.592421	Female

```

Replacing all values in a given column:

Replaced COLUMN 'D' with all 5

```

	A	B	C	D	E	F
2021-03-01	0.073687	0.289288	-1.256938	5	-1.159301	Male
2021-03-02	-0.508095	1.169204	-0.596631	5	-0.227631	Female
2021-03-03	1.299929	1.005249	0.985375	5	1.654088	Male
2021-03-04	-1.024657	-0.858542	-0.104141	5	-0.927653	Female
2021-03-05	-1.550090	-1.572869	-1.488436	5	-2.471808	Male
2021-03-06	-0.161631	-0.498060	-0.864079	5	0.325774	Female
2021-03-07	-0.932733	1.564995	0.494762	5	-0.147785	Male
2021-03-08	-0.721355	0.452387	-0.557525	5	-1.210777	Female
2021-03-09	1.375796	0.225218	-0.277615	5	-0.895532	Male
2021-03-10	-0.153365	0.558645	0.022999	5	-0.592421	Female

```

sorting the values by Column B:

```

	A	B	C	D	E	F
2021-03-05	-1.550090	-1.572869	-1.488436	5	-2.471808	Male
2021-03-04	-1.024657	-0.858542	-0.104141	5	-0.927653	Female
2021-03-06	-0.161631	-0.498060	-0.864079	5	0.325774	Female
2021-03-09	1.375796	0.225218	-0.277615	5	-0.895532	Male
2021-03-01	0.073687	0.289288	-1.256938	5	-1.159301	Male
2021-03-08	-0.721355	0.452387	-0.557525	5	-1.210777	Female

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
2021-03-09 1.375796 0.225218 -0.277615 -0.288386 -0.895532
Adding a sixth column:
      A      B      C      D      E      F
2021-03-01 0.073687 0.289288 -1.256938 0.376190 -1.159301 Male
2021-03-02 -0.508095 1.169204 -0.596631 0.132768 -0.227631 Female
2021-03-03 1.299929 1.005249 0.985375 0.922945 1.654088 Male
2021-03-04 -1.024657 -0.858542 -0.104141 0.335595 -0.927653 Female
2021-03-05 -1.550090 -1.572869 -1.488436 0.346251 -2.471808 Male
2021-03-06 -0.161631 -0.498060 -0.864079 0.687370 0.325774 Female
2021-03-07 -0.932733 1.564995 0.494762 -2.660015 -0.147785 Male
2021-03-08 -0.721355 0.452387 -0.557525 0.747480 -1.210777 Female
2021-03-09 1.375796 0.225218 -0.277615 -0.288386 -0.895532 Male
2021-03-10 -0.153365 0.558645 0.022999 -0.468793 -0.592421 Female
Replacing all values in a given column:
Replaced COLUMN 'D' with all 5
      A      B      C      D      E      F
2021-03-01 0.073687 0.289288 -1.256938 5 -1.159301 Male
2021-03-02 -0.508095 1.169204 -0.596631 5 -0.227631 Female
2021-03-03 1.299929 1.005249 0.985375 5 1.654088 Male
2021-03-04 -1.024657 -0.858542 -0.104141 5 -0.927653 Female
2021-03-05 -1.550090 -1.572869 -1.488436 5 -2.471808 Male
2021-03-06 -0.161631 -0.498060 -0.864079 5 0.325774 Female
2021-03-07 -0.932733 1.564995 0.494762 5 -0.147785 Male
2021-03-08 -0.721355 0.452387 -0.557525 5 -1.210777 Female
2021-03-09 1.375796 0.225218 -0.277615 5 -0.895532 Male
2021-03-10 -0.153365 0.558645 0.022999 5 -0.592421 Female
sorting the values by Column B:
      A      B      C      D      E      F
2021-03-05 -1.550090 -1.572869 -1.488436 5 -2.471808 Male
2021-03-04 -1.024657 -0.858542 -0.104141 5 -0.927653 Female
2021-03-06 -0.161631 -0.498060 -0.864079 5 0.325774 Female
2021-03-09 1.375796 0.225218 -0.277615 5 -0.895532 Male
2021-03-01 0.073687 0.289288 -1.256938 5 -1.159301 Male
2021-03-08 -0.721355 0.452387 -0.557525 5 -1.210777 Female
2021-03-10 -0.153365 0.558645 0.022999 5 -0.592421 Female
2021-03-03 1.299929 1.005249 0.985375 5 1.654088 Male
2021-03-02 -0.508095 1.169204 -0.596631 5 -0.227631 Female
2021-03-07 -0.932733 1.564995 0.494762 5 -0.147785 Male
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 6:

```
#Deleting rows and columns in Dataframe
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(10,3),index=list('abcdefghij'),columns=list('ABC'
))
print("Original Dataframe:\n",df)
df.drop('A',axis=1,inplace=True)
print("Dropping column A:\n",df)
df.drop('e',axis=0,inplace=True)
```

```
print('Dropping 5th row: using index- "e"\n',df)
```

```
ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 Sq6.py
Original DataFrame:
      A      B      C
a  0.580190 -0.801878 -0.002066
b  0.932587 -1.179144 -0.423882
c  0.099776  0.579394 -0.864587
d  0.111814  1.058205  0.792159
e  1.602447 -0.745419 -0.483040
f  0.204692  0.231691 -0.010590
g  2.178834 -0.917409 -0.148154
h -1.251746  0.128746 -0.408536
i  0.497709 -0.587474  0.618973
j -0.275022 -0.021341  0.931371
Dropping column A:
      B      C
a -0.801878 -0.002066
b -1.179144 -0.423882
c  0.579394 -0.864587
d  1.058205  0.792159
e -0.745419 -0.483040
f  0.231691 -0.010590
g -0.917409 -0.148154
h  0.128746 -0.408536
i -0.587474  0.618973
j -0.021341  0.931371
Dropping 5th row: using index- "e"
      B      C
a -0.801878 -0.002066
b -1.179144 -0.423882
c  0.579394 -0.864587
d  1.058205  0.792159
f  0.231691 -0.010590
g -0.917409 -0.148154
h  0.128746 -0.408536
i -0.587474  0.618973
j -0.021341  0.931371
ugcse@prg28:~/Desktop/180905218/lab2$
```

QUESTION 7:

```
#Data frame concatenation
import pandas as pd
import numpy as np
df1=pd.DataFrame(np.random.randn(10,5),index=list('abcdefghij'),columns=list('AB
CDE'))
df2=pd.DataFrame(np.random.randn(10,3),index=list('abcdefghij'),columns=list('AB
C'))
print("Data Frame 1: ",df1.shape)
print("Data Frame 2: ",df2.shape)
```



```

print("Horizontal Concatenation:")
df_new=pd.concat([df1,df2],axis=1)
print('Dimensions of new Horizontal Data Frame: ',df_new.shape)
print("\nVertical Concatenation")
df_vert=pd.concat([df1,df2],axis=0)
print('Dimensions of new Vertical Data Frame: ',df_vert.shape)
print("Absent values substituted by NaN:\n",df_vert)

```

```

ugcse@prg28: ~/Desktop/180905218/lab2
File Edit View Search Terminal Help
ugcse@prg28:~/Desktop/180905218/lab2$ python3 Sq7.py
Data Frame 1: (10, 5)
Data Frame 2: (10, 3)
Horizontal Concatenation:
Dimensions of new Horizontal Data Frame: (10, 8)

Vertical Concatenation
Dimensions of new Vertical Data Frame: (20, 5)
Absent values substituted by NaN:

```

	A	B	C	D	E
a	0.886935	1.849586	1.205058	-0.497427	-1.832625
b	-0.961064	0.030015	-0.579564	-1.512806	1.402907
c	0.070856	0.255828	0.858528	-0.631732	0.240671
d	-1.975602	0.262146	0.016426	-1.232349	-0.287875
e	-0.541443	0.020027	-1.107485	1.445506	0.088210
f	0.173063	-0.760404	-0.780617	2.491436	-0.344016
g	0.173383	1.059868	0.079555	-0.651636	1.995663
h	0.503645	2.126584	-1.173860	0.139021	0.063769
i	-1.275349	0.026076	-0.759286	1.116222	-1.380157
j	1.466375	-0.089503	-1.080667	-1.183724	0.845519
a	-0.336628	1.839650	0.834104	NaN	NaN
b	1.356588	0.001163	0.881264	NaN	NaN
c	1.445752	0.212961	-0.059720	NaN	NaN
d	-1.093471	0.857330	0.223278	NaN	NaN
e	-0.493309	-0.303757	0.013721	NaN	NaN
f	-0.403621	0.172438	-0.009621	NaN	NaN
g	0.361814	0.849854	1.082913	NaN	NaN
h	0.427600	-1.766907	-2.403893	NaN	NaN
i	0.417902	-0.593263	-0.041290	NaN	NaN
j	-1.663787	-0.089512	0.116238	NaN	NaN

```

ugcse@prg28:~/Desktop/180905218/lab2$

```