

Aditi Kathalay

PRN: 1132220011

Python Lab Assignment 1

Q1 Printing Hello World

In [3]:

```
print("Hello World")
```

Hello World

Q2 To get today's Date and time

In [56]:

```
from datetime import datetime  
  
print("Today's date and time is : ",datetime.now())
```

Today's date and time is : 2022-08-25 23:10:47.129554

Q3 To get Python Version

In [1]:

```
import sys  
print("Python version is : ",sys.version)
```

Python version is : 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC
v.1929 64 bit (AMD64)]

Q4 To calculate area of circle, square and triangle

In [16]:

```
def areaCircle(radius):  
    PI = 3.14  
    return PI*(radius*radius);  
rad = float(input("Enter radius : "))  
print("Area of circle is : ",areaCircle(rad))  
  
def areaSquare(side):  
    return side*side;  
sid = float(input("Enter side of square : "))  
print("Area of square is : ",areaSquare(sid))  
  
def areaTriangle(base,height):  
    return 1.5*(base*height);  
bas = float(input("Enter base : "))  
hei = float(input("Enter height : "))  
print("Area of triangle is : ",areaTriangle(bas,hei))
```

```
Enter radius : 22  
Area of circle is : 1519.76  
Enter side of square : 2  
Area of square is : 4.0  
Enter base : 2  
Enter height : 2  
Area of triangle is : 6.0
```

Q5 If Number is positive, negative or zero

In [29]:

```
num = 2  
if num>0:  
    print("Number is positive")  
elif num<0:  
    print("Number is negative")  
else:  
    print("Number is zero")
```

Number is positive

Q6 Convert km to miles

In [26]:

```
km = float(input("Enter kilometers : "))  
if(km!=0):  
    print("Kilometers to miles is : ", km*0.62137119)
```

```
Enter kilometers : 2  
Kilometers to miles is : 1.24274238
```

Q7 Convert celsius to fahrenheit

In [27]:

```
celsius = float(input("Enter celsius : "))  
if(celsius!=0):  
    print("Celsius to Farenheit : ", (celsius*1.8)+32)
```

Enter celsius : 22

Celsius to farenheit : 71.6

Q8 If Number is positive, negative or zero. Ask user to give the input number.

In [28]:

```
num = float(input("Enter number : "))  
if num>0:  
    print("Number is positive")  
elif num<0:  
    print("Number is negative")  
else:  
    print("Number is zero")
```

Enter number : 2

Number is positive

Q9 To find whether a string is palindrome. Ask user to give the input a string.

In [1]:

```
def palindrome(s):  
    return s == s[::-1]  
s = input("Enter string : ")  
ans = palindrome(s)  
  
if ans:  
    print("Yes")  
else:  
    print("No")
```

Enter string : eye

Yes

Q10 If a year is leap year or not

In [42]:

```
year = float(input("Enter year : "))
if (year % 400 == 0):
    print("Year is leap");
elif (year % 100 == 0):
    print("Year is not leap");
elif (year % 4 == 0):
    print("Year is leap");
else:
    print("Year is not leap");
```

Enter year : 2000
Year is leap

Q11 Calculate simple interest

In [43]:

```
principle = float(input("Enter principle amount : "))
rate = float(input("Enter rate of interest : "))
time = float(input("Enter time : "))

SI = (principle*rate*time)/100
print("Simple Interest is : ",SI)
```

Enter principle amount : 2000
Enter rate of interest : 10
Enter time : 1
Simple Interest is : 200.0

Q12 That accepts an integer (n) and computes the value of $n+nn+nnn$.

In [45]:

```
n = float(input("Enter number : "))
ans = n+(n*n)+(n*n*n)
print("Answer is ",ans)
```

Enter number : 1
Answer is 3.0

Q13 To sum three given integers. However, if two values are equal sum will be zero.

In [49]:

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

if(num1 == num2 or num2 == num3 or num1 == num3):
    print("Sum is 0 ")
else:
    print("Sum is : ", num1+num2+num3)
```

```
Enter first number : 2
Enter second number : 3
Enter first number : 2
Sum is 0
```

Q14 Convert in feet to inches, yards, and miles

In [52]:

```
feet = int(input("Input distance in feet: "))
print("The distance in inches is : ", feet * 12)
print("The distance in yards is : ", feet / 3.0)
print("The distance in miles is : ", feet / 5280.0)
```

```
Input distance in feet: 100
The distance in inches is : 1200
The distance in yards is : 33.333333333333336
The distance in miles is : 0.01893939393939394
```

Q15 Convert all units of time into seconds

In [53]:

```
days = int(input("Enter days: ")) * 3600 * 24
hours = int(input("Enter hours: ")) * 3600
minutes = int(input("Enter minutes: ")) * 60
seconds = int(input("Enter seconds: "))

Time = days + hours + minutes + seconds

print("The amounts of seconds", Time)
```

```
Enter days: 5
Enter hours: 10
Enter minutes: 2
Enter seconds: 1
The amounts of seconds 468121
```

Aditi Kathalay

PRN: 1132220011

Python Lab Assignment 2

Q1 Write a python program to construct the following pattern, using a nested for loop

In [7]:

```
for i in range(7):
    for j in range(i):
        print('*', end=" ")
    print()

for i in range(5,0,-1):
    for j in range(i):
        print('*', end=" ")
    print()
```

```
*
**
***
****
*****
*****
*****
****
***
**
*
```

Q2 Write a Python program to find those numbers which are divisible by 7 and multiple

of 5, between 1500 and 2700 (both included)

In [1]:

```
for i in range(1500,2701):  
    if(i % 7 == 0):  
        if(i%5 == 0):  
            print(i)
```

1505
1540
1575
1610
1645
1680
1715
1750
1785
1820
1855
1890
1925
1960
1995
2030
2065
2100
2135
2170
2205
2240
2275
2310
2345
2380
2415
2450
2485
2520
2555
2590
2625
2660
2695

Q3 Write a Python program to count the number of even and odd numbers from a series

of numbers. Sample numbers:

numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)

In [20]:

```
ecount = 0
ocount = 0
for i in range(1,10):
    if(i%2 == 0):
        ecount+=1

    elif(i%2 == 1):
        ocount+=1

print("Even numbers are : ",ecount)
print("Odd numbers are : ",ocount)
```

Even numbers are : 4
Odd numbers are : 5

Q4 Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.

Note: Use 'continue' statement.

In [34]:

```
for i in range(0,7):
    if(i==3 or i==6):
        continue
    print(i)
```

0
1
2
4
5

Q5 Write a Python program to get the Fibonacci series between 0 to 50.

Note: The Fibonacci Sequence is the series of numbers :

0, 1, 1, 2, 3, 5, 8, 13, 21,

In [40]:

```
prev1 = 0
prev = 1

print("Fibonacci series is : ",prev,end=" ")
for i in range(8):
    n=prev+prev1
    prev1=prev
    prev=n
    print(n,end=" ")
```

Fibonacci series is : 1 1 2 3 5 8 13 21 34

Q6 Write a Python program to print alphabet pattern 'A'.

In [30]:

```

for i in range(0,10):
    if i == 0 or i == 5:
        print("*****")
    else:
        print("*       *")

```

```

*****
*       *
*       *
*       *
*       *
*****
*       *
*       *
*       *
*       *

```

Q7 Write a program to check whether a number is Prime number or not.

In [33]:

```

num=int(input("Enter number : "))
if num > 1:
    for i in range(2, int(num/2)+1):
        if (num % i) == 0:
            print(num, "is not a prime")
            break
    else:
        print(num, "is a prime")
else:
    print(num, " is not prime")

```

```

Enter number : 225
225 is a prime
225 is not a prime

```

Q8 Write a Program to find the factorial of a number

In [44]:

```

n = int (input ("Enter a number : "))
fact = 1
if n >= 1:
    for i in range (1, n+1):
        fact = fact * i
print("Factorial of the given number is : ", fact)

```

```

Enter a number : 0
Factorial of the given number is : 1

```

Q9 Write a program to display multiplication table of any number entered by user.

In [10]:

```
n = 11
for i in range(1, 11):
    print(n, 'x', i, '=', n*i)
```

```
11 x 1 = 11
11 x 2 = 22
11 x 3 = 33
11 x 4 = 44
11 x 5 = 55
11 x 6 = 66
11 x 7 = 77
11 x 8 = 88
11 x 9 = 99
11 x 10 = 110
```

Q10 Write a program to find Armstrong number within an interval given by user.

In [2]:

```
n = int(input("Enter a number: "))
sum = 0
temp = n
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10
if n == sum:
    print(n, "is an Armstrong number")
else:
    print(n, "is not an Armstrong number")
```

Enter a number: 455

455 is not an Armstrong number

Q11 Write a program to find factors of a number.

In [39]:

```
x = int(input("Enter number : "))
print("Factors are : ", end=" ")
for i in range(1, x+1):
    if(x % i == 0):
        print(i, end=" ")
```

Enter number : 4

Factors are : 1 2 4

Q12 Write a program to find LCM and HCF.

In [8]:

```
a = int(input("Enter first number : "))
b = int(input("Enter second number : "))
for i in range(2,a+1):
    if(a % i == 0 and b % i == 0):
        HCF=i
        LCM=int((a*b)/HCF)
print("HCF is : ",HCF)
print("LCM is : ",LCM)
```

```
Enter first number : 78
Enter second number : 104
HCF is : 26
LCM is : 312
```

Aditi Kathalay

PRN: 1132220011

Python Lab Assignment 3

A) List

Q1 Write a Python program to sum all the items in a list.

In [1]:

```
list = [1,2,3,4,5]
sum = 0
for key in range(0,len(list)):
    sum += list[key]
print("Sum of elements in list is : ",sum)
```

Sum of elements in list is : 15

Q2 Write a Python program to get the largest and smallest number from a list

In [4]:

```
list = [1,2,3,4]
print("Smallest number is :",min(list))
print("Largest number is :",max(list))
```

Smallest number is : 1

Largest number is : 4

Q3 Write a Python program to check a list is empty or not.

In [7]:

```
list={1,2,3,4}
if len(list)==0:
    print("List is empty")
else:
    print("List is not empty")
```

List is not empty

Q4 Write a Python program to clone or copy a list

In [15]:

```
list = [1,2,3,4]
new_list = list.copy()
print(new_list)
```

[1, 2, 3, 4]

Q5 Write a Python program to print a specified list after removing the 0th, 4th and

5th elements.

In [13]:

```
list =[1,2,3,4,5,6,7,8,9]
del list[0]
del list[4]
del list[5]
print(list)
```

[2, 3, 4, 5, 7, 9]

In [47]:

```
list1 = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
del(list1[0])
del(list1[3])
del(list1[3])
print(list1)
```

['Green', 'White', 'Black']

Q6 Write a Python program access the index of a list

In [50]:

```
lists = [1,2,3,4,5]

for i in range(len(lists)):
    print(i, " : ", lists[i])
```

```
0 : 1
1 : 2
2 : 3
3 : 4
4 : 5
```

Q7 Write a Python program to append a list to the second list.

In [18]:

```
list1 = [1,2,3,4]
list2 = [5,6,7,8]
new_list = list1+list2
print(new_list)
```

[1, 2, 3, 4, 5, 6, 7, 8]

Q8 Write a Python program to select an item randomly from a list.

In [3]:

```
import random
list = [1,2,3,4,5]
print("Randomly selected element is :", random.choice(list))
```

Randomly selected element is : 1

Q9 Write a Python program to get unique values from a list

In [8]:

```
list = [1, 2, 1, 3, 4, 4]
def unique(list):
    unique_ = []
    for x in list:
        # check if exists in unique_list or not
        if x not in unique_:
            unique_.append(x)
    # print list
    for x in unique_:
        print(x)
print("The unique values from 1st list is : ")
unique(list)
```

```
The unique values from 1st list is :
10
20
30
40
```

Q10 Write a Python program to get the second largest and second smallest number**from a list**

In [18]:

```
def find_len(list1):
    length = len(list1)
    list1.sort()
    print("Sorted list is : ",list1)
    print("Second Largest element is:", list1[length-2])
    print("Second Smallest element is:", list1[1])
list1 = [1, 4, 2, 22, 31, 78, 90, 36, 40]
Largest = find_len(list1)
```

```
Sorted list is : [1, 2, 4, 22, 31, 36, 40, 78, 90]
Second Largest element is: 78
Second Smallest element is: 2
```

Q11 Write a Python program to remove duplicates from a list

In [20]:

```
def Remove(list_dup):
    final_list = []
    for num in list_dup:
        if num not in final_list:
            final_list.append(num)
    return final_list
list_dup = [12, 13, 10, 20, 12, 14, 13, 4]
print("List after removal of duplicate items is : ",Remove(list_dup))
```

```
List after removal of duplicate items is : [12, 13, 10, 20, 14, 4]
```

In []:

B) Tuple**Q1 Write a Python program to create a tuple**

In [22]:

```
tuple = (1,2,3)
print("Tuple is : ",tuple)
```

```
Tuple is : (1, 2, 3)
```

Q2 Write a Python program to create a tuple with different data types

In [24]:

```
tuple1 = ("String", True, 1.2,9)
print(tuple1)
```

```
('String', True, 1.2, 9)
```

Q3 Write a Python program to create a tuple with numbers and print one item

In [26]:

```
tuple2 = (1,2,3)
print(tuple2[2])
```

```
3
```

Q4 Write a Python program to add an item in a tuple

In [51]:

```
append = tuple2 +(34,)
print(append)
```

```
(1, 2, 3, 34)
```

Q5 Write a Python program to get the 4th element and 4th element from last of a tuple.

In [30]:

```
tuple4 = (1,2,3,4,5,6,7,8,9,10,11,12)
print("4th element is : ",tuple4[3])
print("4th last element is : ",tuple4[-4])
```

```
4th element is : 4
4th last element is : 9
```

Q6 Write a Python program to check whether an element exists within a tuple.

In [34]:

```
tuple5 = (1,2,3,4,5,6,7,8)
x = int(input("Enter element to check in tuple: "))
if x in tuple5:
    print("Element exists")
else:
    print("Element doesn't exists")
```

```
Enter element to check in tuple: 5
Element exists
```

Q7 Write a Python program to convert a list to a tuple.

In [2]:

```
sample_list = [1,2,3]
tuple1 = (*sample_list,)
print(tuple1)
print(type(tuple1))
```

```
(1, 2, 3)
<class 'tuple'>
```

Q8 Write a Python program to slice a tuple.

In [39]:

```
tuple6 = (1,2,3,4,5)
print("Sliced tuple is: ",tuple6[1:3])
```

```
Sliced tuple is : (2, 3)
```

Q9 Write a Python program to find the index of an item of a tuple.

In [40]:

```
tuple7 = (1,2,3,4,5,6,7,8)
print("Index of 2 is: ",tuple7.index(2))
```

```
Index of 2 is : 1
```

Q10 Write a Python program to find the length of a tuple.

In [41]:

```
print("Length of tuple is : ",len(tuple7))
```

```
Length of tuple is : 8
```

Q11 Write a Python program to sort a tuple by its float element

In [4]:

```
stationery = [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]
print( sorted(stationery, key=lambda x: float(x[1]), reverse=True))

[('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]
```

C) Strings**Q1 Write a Python program to count the number of characters (character frequency) in a string**

In [5]:

```
def frequency(str1):
    dict = {}
    for n in str1:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict
print(frequency("MIT World Peace University"))

{'M': 1, 'I': 1, 'T': 1, ' ': 3, 'W': 1, 'o': 1, 'r': 2, 'l': 1, 'd': 1, 'P': 1, 'e': 3, 'a': 1, 'c': 1, 'U': 1, 'n': 1, 'i': 2, 'v': 1, 's': 1, 't': 1, 'y': 1}
```

Q2 Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

In [11]:

```
a = (input("Enter first character: "))
b = (input("Enter second character: "))
def chars_swap(a, b):
    new_a = b[:2] + a[2:]
    new_b = a[:2] + b[2:]

    return new_a + ' ' + new_b
print("Characters after swapping strings are: ",chars_swap(a,b))
```

```
Enter first character: hello
Enter second character: world
Characters after swapping strings are:  wollo herld
```


Q3 Write a Python program to add 'ing' at the end of a given string (length should be at least 3). 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 unchanged.

In [15]:

```
str1 = input("Enter String: ")
def add_in_string(str1):
    length = len(str1)

    if length > 2:
        if str1[-3:] == 'ing':
            str1 += 'ly'
        else:
            str1 += 'ing'

    return str1
print(add_in_string(str1))
```

Enter String: hello
helloing

Q4 Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string

In [20]:

```
def appearance(string):
    snot = string.find('not')
    spoor = string.find('poor')

    if spoor > snot and snot > 0 and spoor > 0:
        string = string.replace(string[snot:(spoor+4)], 'good')
        return string
    else:
        return string
print(appearance('The weather is not so poor'))
print(appearance('The weather is poor'))
```

The weather is good
The weather is poor

Aditi Kathalay

PRN: 1132220011

Python Lab Assignment 4

A) Set

Q1 Write a Python program to create a set

In [2]:

```
set1 = {1,2,3,4,5}
print("Set is :", set1)
```

Set is : {1, 2, 3, 4, 5}

Q2 Write a Python program to iterate over sets

In [5]:

```
set2 = set([0, 1, 2, 3, 4, 5])
for n in set2:
    print(n, end=' ')
```

0 1 2 3 4 5

Q3 Write a Python program to add member(s) in a set

In [14]:

```
stationery = {"Pencil","Pen"}
print(stationery)
stationery.update(["Scale", "Eraser"])
print("After adding multiple items : ", stationery)
```

{'Pen', 'Pencil'}
After adding multiple items : {'Scale', 'Eraser', 'Pen', 'Pencil'}

Q4 Write a Python program to remove item(s) from set

In [21]:

```
set3 = set([0, 1, 2, 3, 4, 5])
print("Original set elements: ",set3)
set3.discard(0)
print("\nAfter removing 0 from the said set: ",set3)
```

Original set elements: {0, 1, 2, 3, 4, 5}

After removing 0 from the said set: {1, 2, 3, 4, 5}

Q5 Write a Python program to remove an item from a set if it is present in the set

In []:

```
set4 = set([0, 1, 2, 3, 4, 5])
print("Original set elements: ",set4)
element = int(input("Enter value to be removed: "))
if (element in set4):
    set4.discard(element)
    print("After removing element set is: ",set4)
else:
    print("Element not present in set")
```

Q6 Write a Python program to create an intersection of sets

In [2]:

```
set5 = set([1,2,3])
set6 = set([1,3,4,5,6])
print("The sets are: ",set5,set6)
set7 = set5 & set6
print("The intersection of sets is: ",set7)
```

The sets are: {1, 2, 3} {1, 3, 4, 5, 6}
The intersection of sets is: {1, 3}

Q7 Write a Python program to create a union of sets

In [5]:

```
set5 = set([1,2,3])
set6 = set([4,5,6])
print("The sets are: ",set5,set6)
set7 = set5.union(set6)
print("The union of sets is: ",set7)
```

The sets are: {1, 2, 3} {4, 5, 6}
The union of sets is: {1, 2, 3, 4, 5, 6}

Q8 Write a Python program to create set difference

In [12]:

```
set8 = set([1, 1, 2, 3, 4, 5])
set9 = set([1, 5, 6, 7, 8, 9])
print("\nOriginal sets are: ",set8,set9)
output1 = set8.difference(set9)
print("\nDifference of set8 - set9 is: ",output1)
output2 = set9.difference(set8)
print("\nDifference of set9 - set8 is: ",output2)
```

Original sets are: {1, 2, 3, 4, 5} {1, 5, 6, 7, 8, 9}

Difference of set8 - set9 is: {2, 3, 4}

Difference of set9 - set8 is: {8, 9, 6, 7}

Q9 Write a Python program to create a symmetric difference.

In [11]:

```
set8 = set([1, 1, 2, 3, 4, 5])
set9 = set([1, 5, 6, 7, 8, 9])
print("\nOriginal sets are: ",set8,set9)
output1 = set8.symmetric_difference(set9)
print("\nSymmetric difference of set8 - set9 is: ",output1)
output2 = set9.symmetric_difference(set8)
print("\nSymmetric difference of set9 - set8 is: ",output2)
```

Original sets are: {1, 2, 3, 4, 5} {1, 5, 6, 7, 8, 9}

Symmetric difference of set8 - set9 is: {2, 3, 4, 6, 7, 8, 9}

Symmetric difference of set9 - set8 is: {2, 3, 4, 6, 7, 8, 9}

Q10 Write a Python program to issubset and issuperset.

In [13]:

```
SetA = {4, 1, 3, 5}
SetB = {6, 0, 4, 1, 5, 0, 3, 5}

print("SetA.issuperset(SetB) : ", SetA.issuperset(SetB))
print("SetB.issuperset(SetA) : ", SetB.issuperset(SetA))
```

SetA.issuperset(SetB) : False
SetB.issuperset(SetA) : True

Q11 Write a Python program to create a shallow copy of sets

In [20]:

```
A = set(["Green", "Red"])
C = A.copy()
print(C)
```

```
{'Green', 'Red'}
```

Q12 Write a Python program to clear a set.

In [23]:

```
set10 = {1, 2, 3, 4}
print("Set is: ",set10)
set10.clear()
print(set10)
```

```
Set is: {1, 2, 3, 4}
set()
```

B) Dictionary**Q1 Write a Python script to sort (ascending and descending) a dictionary by value**

In [27]:

```
dict1={'Pencil':20,'Pen':2,'Eraser':1,'Scales':3}
l=list(dict1.items())
l.sort()
print('Ascending order is',l)
l=list(dict1.items())
l.sort(reverse=True)
print('Descending order is',l)
dict2=dict(l)
print("Dictionary",dict2)
```

```
Ascending order is [('Eraser', 1), ('Pen', 2), ('Pencil', 20), ('Scales', 3)]
Descending order is [('Scales', 3), ('Pencil', 20), ('Pen', 2), ('Eraser', 1)]
```

```
-----
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_9548\1742031527.py in <module>
      6 l.sort(reverse=True)
      7 print('Descending order is',l)
----> 8 dict2=dict(l)
      9 print("Dictionary",dict2)
```

```
TypeError: 'dict' object is not callable
```

Q2 Write a Python script to add a key to a dictionary

In [29]:

```
dict3 = {1:10, 2:20}
print("Dictionary is: ",dict3)
dict3.update({3:30})
print("Dictionary after updating is: ",dict3)
```

```
Dictionary is: {1: 10, 2: 20}
Dictionary after updating is: {1: 10, 2: 20, 3: 30}
```

Q3 Write a Python script to concatenate following dictionaries to create a new one

In [31]:

```
dict4={1:10, 2:20}
dict5={3:30, 4:40}
dict6={5:50,6:60}
dict7 = {}
for d in (dict4, dict5, dict6):
    dict7.update(d)
print(dict7)
```

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Q4 Write a Python script to check whether a given key already exists in a dictionary

In [32]:

```
dict8 = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
def key_present(x):
    if x in dict8:
        print('Key is present in the dictionary')
    else:
        print('Key is not present in the dictionary')
key_present(5)
key_present(9)
```

Key is present in the dictionary
Key is not present in the dictionary

Q5 Write a Python program to iterate over dictionaries using for loops.

In [37]:

```
dict9 = {'fruit?': 'apple', 'vegetable?': 'spinach', 'hotel?': 'trivago'}
for key, value in dict9.items():
    print(key, value)
```

fruit? apple
vegetable? spinach
hotel? trivago

Q6 Write a Python script to generate and print a dictionary that contains a number**(between 1 and n) in the form (x, x*x).**

In [1]:

```
n=int(input("Enter a number: "))
dict10={x:x*x for x in range(1,n+1)}
print("Dictionary is: ",dict10)
```

Enter a number: 3
Dictionary is: {1: 1, 2: 4, 3: 9}

Q7 Write a Python script to print a dictionary where the keys are numbers between 1**and 15 (both included) and the values are square of keys**

In [2]:

```
dict11=dict()
for x in range(1,16):
    dict11[x]=x**2
print(dict11)
```

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

Q8 Write a Python script to merge two Python dictionaries

In [6]:

```
dict12 = {'a': 100, 'b': 200}
dict13 = {'x': 300, 'y': 200}
dict14 = dict12.copy()
dict14.update(dict13)
print("First dictionary is: ",dict12)
print("Second dictionary is: ",dict13)
print("Merged dictionary is: ",dict14)
```

First dictionary is: {'a': 100, 'b': 200}
Second dictionary is: {'a': 100, 'b': 200}
Merged dictionary is: {'a': 100, 'b': 200, 'x': 300, 'y': 200}

Q9 Write a Python program to iterate over dictionaries using for loops

In [7]:

```
dict15 = {'Pencil': 1, 'Pen': 2, 'Scale': 3}
for c, value in dict15.items():
    print(c, ': ', dict15[c])
```

```
Pencil : 1
Pen : 2
Scale : 3
```

Q10 Write a Python program to sum all the items in a dictionary

In [10]:

```
def val_sum(dict):
    return sum(dict.values())
dict = {'a': 10, 'b': 20, 'c': 30}
print("Sum is: ", val_sum(dict))
```

```
Sum is: 60
```

Q11 Write a Python program to remove a key from a dictionary

In [12]:

```
dict16 = {'Pencil':10,'Pen':2,'Scale':3,'Eraser':4}
print(dict16)
if 'Pencil' in dict16:
    del dict16['Pencil']
print(dict16)
```

```
{'Pencil': 10, 'Pen': 2, 'Scale': 3, 'Eraser': 4}
{'Pen': 2, 'Scale': 3, 'Eraser': 4}
```

Q12 Write a Python program to get the maximum and minimum value in a dictionary

In [14]:

```
dict17 = {'Pencil':10, 'Pen':20, 'Scale': 30}
key_max = max(dict17.keys(), key=(lambda k: dict17[k]))
key_min = min(dict17.keys(), key=(lambda k: dict17[k]))
print('Maximum Value is: ',dict17[key_max])
print('Minimum Value is: ',dict17[key_min])
```

```
Maximum Value is: 30
Minimum Value is: 10
```

Q13 Write a Python program to combine two dictionary adding values for common keys

In [21]:

```
dict18 = {'good': 6, 'and': 25, 'Do': 9, 'it': 10 }
dict19 = {'in': 100, 'throw': 200, 'the': 300, 'river': 550}
c = {i: dict18.get(i, 0) + dict19.get(i, 0)
     for i in set(dict18).union(dict19)}
print(c)
```

```
{'good': 6, 'Do': 9, 'and': 25, 'river': 550, 'throw': 200, 'in': 100, 'it': 10, 'the': 300}
```

Q14 Write a Python program to print all unique values in a dictionary

In [23]:

```
dict20 = [{"A1": "10"}, {"A": "20"}, {"A2": "10"}, {"A2": "30"}, {"A3": "30"}, {"A2": "20"}, {"A3": "50"}]
print("Dictionary is: ",dict20)
u_value = set( val for dic in dict20 for val in dic.values())
print("Unique Values: ",u_value)
```

```
Dictionary is: [{'A1': '10'}, {'A': '20'}, {'A2': '10'}, {'A2': '30'}, {'A3': '30'}, {'A2': '20'}, {'A3': '50'}]
Unique Values: {'20', '10', '50', '30'}
```

Q15 Write a Python program to get the top three items in a shop

In [30]:

```
from heapq import nlargest
from operator import itemgetter
dict21 = {'Pencils': 100, 'Pens': 32, 'Erasers': 60, 'Scales': 70 }
for name, value in nlargest(3, dict21.items(), key=itemgetter(1)):
    print(name, value)
```

```
Pencils 100
Scales 70
Erasers 60
```

Aditi Kathalay

PRN: 1132220011

Python Lab Assignment 5

Q1 Write a Python function to find the Max of three numbers

In [2]:

```
def max_of_three():  
    a = int(input("Enter 1st number: "))  
    b = int(input("Enter 2nd number: "))  
    c = int(input("Enter 3rd number: "))  
    print("Maximum of three numbers is: ",max(a,b,c))  
max_of_three()
```

```
Enter 1st number: 1  
Enter 2nd number: 2  
Enter 3rd number: 3  
Maximum of three numbers is: 3
```

Q2 Write a Python function to sum all the numbers in a list. Sample List : (8, 2, 3, 0, 7)

Expected Output : 20

In [6]:

```
def sum_of_numbers():  
    sum=0  
    list1 = [8,2,3,0,7]  
    for val in list1:  
        sum+=val  
    print("Sum of numbers in list is:",sum)  
sum_of_numbers()
```

```
Sum of numbers in list is: 20
```

Q3 Write a Python function to multiply all the numbers in a list. Sample List : (8, 2, 3, -1,7)

In [10]:

```
def mul_of_numbers():  
    mul=1  
    list1 = [8,2,3,-1,7]  
    for val in list1:  
        mul*=val  
    print("Sum of numbers in list is:",mul)  
mul_of_numbers()
```

```
Sum of numbers in list is: -336
```

Q4 Write a Python function that accepts a string and calculate the number of upper case

letters and lower case letters. Go to the editor

In [7]:

```
def string():  
    word = input("Enter string: ")  
    count_upper = 0  
    count_lower = 0  
    for letter in word:  
        if (letter.islower()):  
            count_lower+=1  
        else:  
            count_upper+=1  
    print("Upper Case letters are:",count_upper,"and Lower Case letters are:",count_lower)  
string()
```

```
Enter string: Hello  
Upper Case letters are: 1 and Lower Case letters are: 4
```


Q5 Write a Python function that checks whether a passed string is palindrome or not

In []:

```
def palindrome():  
    word = input("Enter string: ")
```

Q6 Write a Python function that prints out the first n rows of Pascal's triangle

In [3]:

```
def pascal_triangle(n):  
    row = [1]  
    b = [0]  
    for a in range(max(n,0)):  
        print(row)  
        row=[l+r for l,r in zip(row+b, b+row)]  
    return n>=1  
pascal_triangle(4)
```

```
[1]  
[1, 1]  
[1, 2, 1]  
[1, 3, 3, 1]
```

Out[3]:

True

Q7 Write a Python program that accepts a hyphen-separated sequence of words as input**and prints the words in a hyphen-separated sequence after sorting them****alphabetically**

In [1]:

```
word=[n for n in input().split('-')]  
word.sort()  
print('-'.join(word))
```

```
h-e-l-l-o  
e-h-l-l-o
```

Q8 Write a Python function to create and print a list where the values are square of**numbers between 1 and 30 (both included).**

In [2]:

```
def square_list():  
    l = list()  
    for i in range(1,30):  
        l.append(i**2)  
    print(l)  
square_list()
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729,  
784, 841]
```

In []:

Aditi Kathalay

PRN: 1132220011

Python Lab Assignment 6

Q1 Write a NumPy program to print the NumPy version in your system

In [3]:

```
import numpy as np
print("Numpy version is: ", np.__version__)
Numpy version is: 1.22.1
```

Q2 Write a NumPy program to convert a list of numeric value into a one-dimensional NumPy array.

In [5]:

```
list1 = [1,2,3,4,5]
print("List is: ", list1)
arr = np.array(list1)
print("One dimensional array is: ")
List is: [1, 2, 3, 4, 5]
One dimensional array is:
```

Q3 Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

In [4]:

```
np.arange(2,11).reshape(3,3)
Out[4]:
array([[ 2,  3,  4],
       [ 5,  6,  7],
       [ 8,  9, 10]])
```

Q5 Write a NumPy program to create a null vector of size 10 and update sixth value to 11

In [6]:

```
arr1 = np.zeros(10)
print("Array before: ", arr1)
arr1[5] = 11
print(arr1)
Array before: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[ 0.  0.  0.  0.  0. 11.  0.  0.  0.  0.]
```

Q6 Write a NumPy program to reverse an array (first element becomes last).

In [7]:

```
arr2 = np.array([1,2,3,4,5])
print(arr2[::-1])
[5 4 3 2 1]
```

Q7 Write a NumPy program to create a 8x8 matrix and fill it with a checkerboard pattern.

In [8]:

```
arr2 = np.zeros((8,8))
for i in range(0,8):
    for j in range(0,8):
        if(i%2==0):
            if(j%2==0):
                arr2[i][j]=1
        else:
            if(j%2!=0):
                arr2[i][j]=1

print(arr2)

[[1.  0.  1.  0.  1.  0.  1.  0.]
 [0.  1.  0.  1.  0.  1.  0.  1.]
 [1.  0.  1.  0.  1.  0.  1.  0.]
 [0.  1.  0.  1.  0.  1.  0.  1.]
 [1.  0.  1.  0.  1.  0.  1.  0.]
 [0.  1.  0.  1.  0.  1.  0.  1.]
 [1.  0.  1.  0.  1.  0.  1.  0.]
 [0.  1.  0.  1.  0.  1.  0.  1.]]
```

Q8 Write a NumPy program to convert the values of Centigrade degrees into Fahrenheit degrees and vice versa. Values are stored into a NumPy array.

In [9]:

```
arr3 = np.array([0,16,23,28,32])
print((np.round(9*arr3/5 + 32),2))

(array([32., 61., 73., 82., 90.]), 2)
```

Q9 Write a NumPy program to find common values between two arrays.

In [10]:

```
arr4 = np.array([1, 2, 3, 4, 5])
print("First array is: ",arr4)
arr5 = [0, 3, 4]
print("Second array is: ",arr5)
print("Common values between two arrays are: ")
print(np.intersect1d(arr4, arr5))

First array is: [1 2 3 4 5]
Second array is: [0, 3, 4]
Common values between two arrays are:
[3 4]
```

Q10 Write a NumPy program to test whether all elements in an array evaluate to True.(Non Zero is True and Zero is False)

In [11]:

```
arr6 = np.array([1, 2, 3, 4])
print("Array is: ",arr6)
print("All elements are: ",np.all(arr6))
arr7 = np.array([0, 1, 2, 3])
print("Array is: ",arr7)
print("Elements in array are: ",np.all(arr7))

Array is: [1 2 3 4]
All elements are: True
Array is: [0 1 2 3]
Elements in array are: False
```

Aditi Kathalay

PRN: 1132220011 

Python Lab Assignment 7

Q1 Write a Pandas program to get the powers of an array values element wise.

Note: First array elements raised to powers from second array

In [4]:

```
import pandas as pd
import numpy as np

array1 = np.array([1,2,3,4,5])
list1 = []
for a in array1:
    list1.append(a*a)
array2 = np.array(list1)
data = pd.DataFrame({'Value':array1, 'Square':array2})
print(data)
```

	Value	Square
0	1	1
1	2	4
2	3	9
3	4	16
4	5	25

Q2 Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels

In [2]:

```
import pandas as pd
import numpy as np

data_frame = {'name': ['Janhavi', 'Falguni', 'Pradnya', 'Shruti', 'Ankita', 'Gloria', 'Mansi', 'Jyoti', 'Angela', 'Shrutika'],
              'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
              'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
              'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(data_frame , index=labels)
print(df)
```

	name	score	attempts	qualify
a	Janhavi	12.5	1	yes
b	Falguni	9.0	3	no
c	Pradnya	16.5	2	yes
d	Shruti	NaN	3	no
e	Ankita	9.0	2	no
f	Gloria	20.0	3	yes
g	Mansi	14.5	1	yes
h	Jyoti	NaN	1	no
i	Angela	8.0	2	no
j	Shrutika	19.0	1	yes

Q3 Write a Pandas program to display a summary of the basic information**about a specified DataFrame and its data**

In [4]:

```
df = pd.DataFrame(data_frame , index=labels)
print("Summary of the basic information: ")
print(df.info())
```

```
Summary of the basic information:
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   name         10 non-null      object
1   score        8 non-null       float64
2   attempts     10 non-null      int64
3   qualify      10 non-null      object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None
```

Q4 Write a Pandas program to get the first 3 rows of a given DataFrame

In [5]:

```
df = pd.DataFrame(data_frame , index=labels)
print("first 3 rows of a given DataFrame: ")
print(df.head(3))
```

```
first 3 rows of a given DataFrame:
   name  score  attempts  qualify
a  Janhavi  12.5         1     yes
b  Falguni   9.0         3     no
c  Pradnya  16.5         2     yes
```

Q5 Write a Pandas program to select the specified columns and rows from a**given data frame**

In [3]:

```
data = {'score': [12.5, 20, 16.5,14], 'name': ['earth', 'mars', 'jupyter', 'saturn']}
df = pd.DataFrame(data)
print("Select specific columns and rows:")
print(df.iloc[[0,1,2]])
```

```
Select specific columns and rows:
   score  name
0   12.5  earth
1   20.0   mars
2   16.5 jupyter
```

Q6 Write a Pandas program to count the number of rows and columns of a**DataFrame.**

In [6]:

```
df = pd.DataFrame(data_frame , index=labels)
total_rows=len(df.axes[0])
total_cols=len(df.axes[1])
print("Number of Rows: "+str(total_rows))
print("Number of Columns: "+str(total_cols))
```

```
Number of Rows: 10
Number of Columns: 4
```

Q7 Write a Pandas program to select the rows where the score is missing, i.e.

is NaN.

In [8]:

```
df = pd.DataFrame(data_frame , index=labels)
print("Rows where score is missing:")
print(df[df['score'].isnull()])
```

```
Rows where score is missing:
      name  score  attempts  qualify
d Shruti   NaN      3        no
h Jyoti   NaN      1        no
```

Q8 Write a Pandas program to select the rows where number of attempts in

the examination is less than 2 and score greater than 15

In [2]:

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
             'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
             'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
             'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Number of attempts in the examination is less than 2 and score greater than 15 :")
print(df[(df['attempts'] < 2) & (df['score'] > 15)])
```

```
Number of attempts in the examination is less than 2 and score greater than 15 :
      name  score  attempts  qualify
j Jonas   19.0      1        yes
```

Q9 Write a Pandas program to calculate the sum of the examination attempts

by the students

In [11]:

```
print("\nSum of the examination attempts by the students:")
print(df['attempts'].sum())
```

```
Sum of the examination attempts by the students:
19
```

Q10 Write a Pandas program to calculate the mean score for each different student in DataFrame

In [12]:

```
print("\nMean score for each different student in data frame:")
print(df['score'].mean())
```

```
Mean score for each different student in data frame:
13.5625
```

Q11 Write a Pandas program to append a new row 'k' to data frame with given**values for each column. Now delete the new row and return the original****DataFrame.**

In [13]:

```
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nAppend a new row:")
df.loc['k'] = [1, 'Suresh', 'yes', 15.5]
print("Print all records after insert a new record:")
print(df)
print("\nDelete the new row and display the original rows:")
df = df.drop('k')
print(df)
```

Original rows:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

Append a new row:

Print all records after insert a new record:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes
k	1 Suresh		yes	15.5

Delete the new row and display the original rows:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

Q12 Write a Pandas program to sort the DataFrame first by 'name' in**descending order, then by 'score' in ascending order.**

In [14]:

```
df.sort_values(by=['name', 'score'], ascending=[False, True])
print("Sort the data frame first by 'name' in descending order, then by 'score' in ascending order:")
print(df)
```

Sort the data frame first by 'name' in descending order, then by 'score' in ascending order:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

Q13 Write a Pandas program to replace the 'qualify' column contains the**values 'yes' and 'no' with True and False**

In [15]:

```
print("\nReplace the 'qualify' column contains the values 'yes' and 'no' with True and False:")
df['qualify'] = df['qualify'].map({'yes': True, 'no': False})
print(df)
```

Replace the 'qualify' column contains the values 'yes' and 'no' with True and False:

	name	score	attempts	qualify
a	Anastasia	12.5	1	True
b	Dima	9.0	3	False
c	Katherine	16.5	2	True
d	James	NaN	3	False
e	Emily	9.0	2	False
f	Michael	20.0	3	True
g	Matthew	14.5	1	True
h	Laura	NaN	1	False
i	Kevin	8.0	2	False
j	Jonas	19.0	1	True

Q14 Write a Pandas program to change the name 'James' to 'Suresh' in name**column of the DataFrame**

In [3]:

```
print("\nChange the name 'James' to 'Suresh':")
df['name'] = df['name'].replace('James', 'Suresh')
print(df)
```

Change the name 'James' to 'Suresh':

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	Suresh	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

Q15 Write a Pandas program to iterate over rows in a DataFrame

In [4]:

```
import pandas as pd
import numpy as np
exam_data1 = [{'name': 'Anastasia', 'score': 12.5}, {'name': 'Dima', 'score': 9}, {'name': 'Katherine', 'score': 16.5}]
df = pd.DataFrame(exam_data1)
for index, row in df.iterrows():
    print(row['name'], row['score'])
```

Anastasia 12.5
Dima 9.0
Katherine 16.5

Q16 Write a Pandas program to add one row in an existing DataFrame

In [5]:

```
import pandas as pd
import numpy as np
d = {'col1': [1, 4, 3, 4, 5], 'col2': [4, 5, 6, 7, 8], 'col3': [7, 8, 9, 0, 1]}
df = pd.DataFrame(data=d)
print("Original DataFrame")
print(df)
print('After add one row:')
df2 = {'col1': 10, 'col2': 11, 'col3': 12}
df = df.append(df2, ignore_index=True)
print(df)
```

Original DataFrame

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1

After add one row:

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1
5	10	11	12

C:\Users\DELL\AppData\Local\Temp\ipykernel_14020\3932112541.py:9: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
df = df.append(df2, ignore_index=True)
```

Q17 Write a Pandas program to change the order of a DataFrame columns.

In [7]:

```
print('After altering col1 and col3')
df = df[['col3', 'col2', 'col1']]
print(df)
```

After altering col1 and col3

	col3	col2	col1
0	7	4	1
1	8	5	4
2	9	6	3
3	0	7	4
4	1	8	5
5	12	11	10

Q18 Write a Pandas program to select rows from a given DataFrame based on**values in some columns**

In [8]:

```
print('Rows for column1 value == 4')
print(df.loc[df['col1'] == 4])
```

Rows for column1 value == 4

	col3	col2	col1
1	8	5	4
3	0	7	4

Q19 Write a Pandas program to rename columns of a given DataFrame

In [9]:

```
df.columns = ['Column1', 'Column2', 'Column3']
df = df.rename(columns={'col1': 'Column1', 'col2': 'Column2', 'col3': 'Column3'})
print("New DataFrame after renaming columns:")
print(df)
```

New DataFrame after renaming columns:

	Column1	Column2	Column3
0	7	4	1
1	8	5	4
2	9	6	3
3	0	7	4
4	1	8	5
5	12	11	10

Q20 Write a Pandas program to get list from DataFrame column headers

In [10]:

```
import pandas as pd
import numpy as np
exam_data2 = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
              'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
              'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
              'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data2, index=labels)
print(list(df.columns.values))
```

['name', 'score', 'attempts', 'qualify']

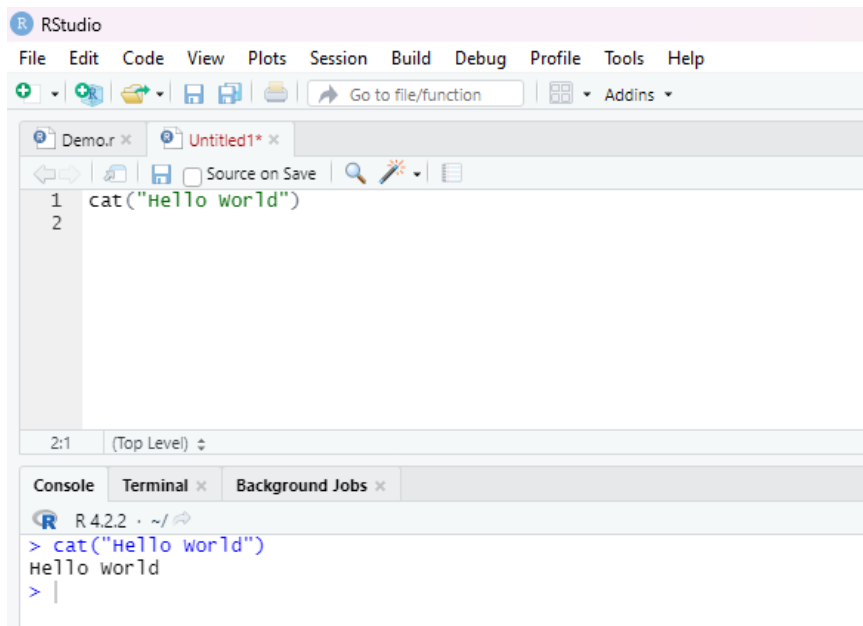
In []:

R Programming Assignment

Name: Aditi Chetan Kathalay PRN: 1132220011

Write a program in R for following

1. Printing Hello World



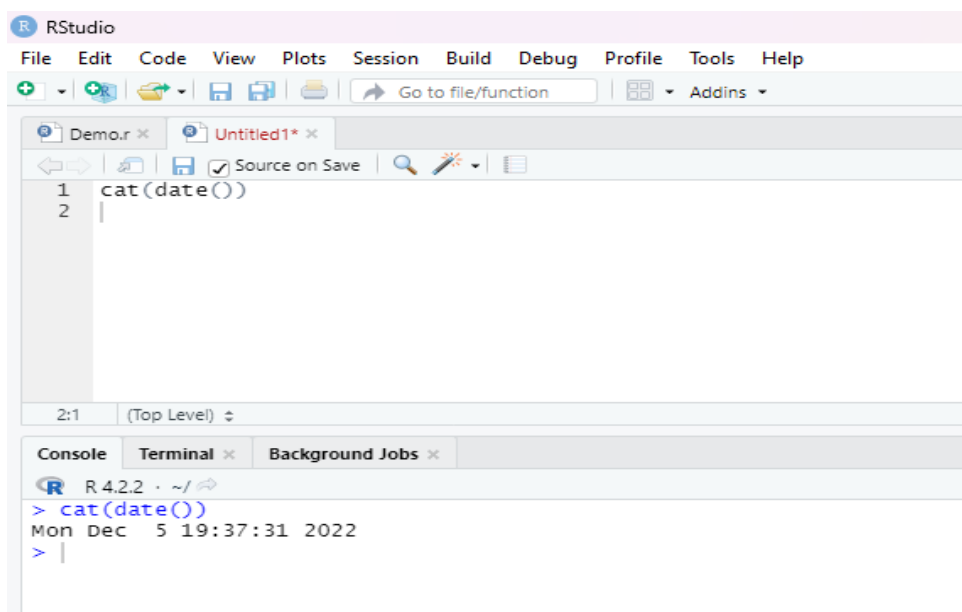
The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 cat("Hello world")
2
```

The console output shows the result of the command:

```
> cat("Hello world")
Hello world
>
```

2. To get today's date and current time



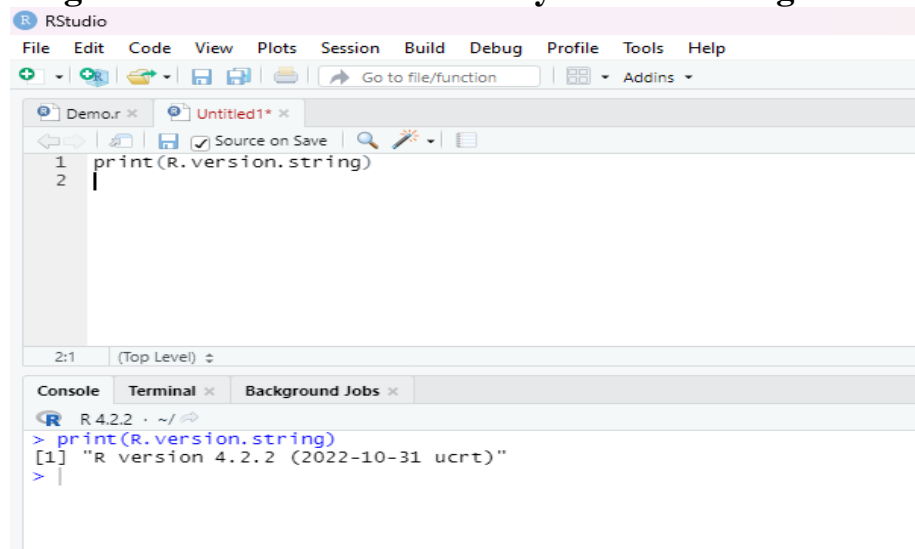
The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 cat(date())
2
```

The console output shows the result of the command:

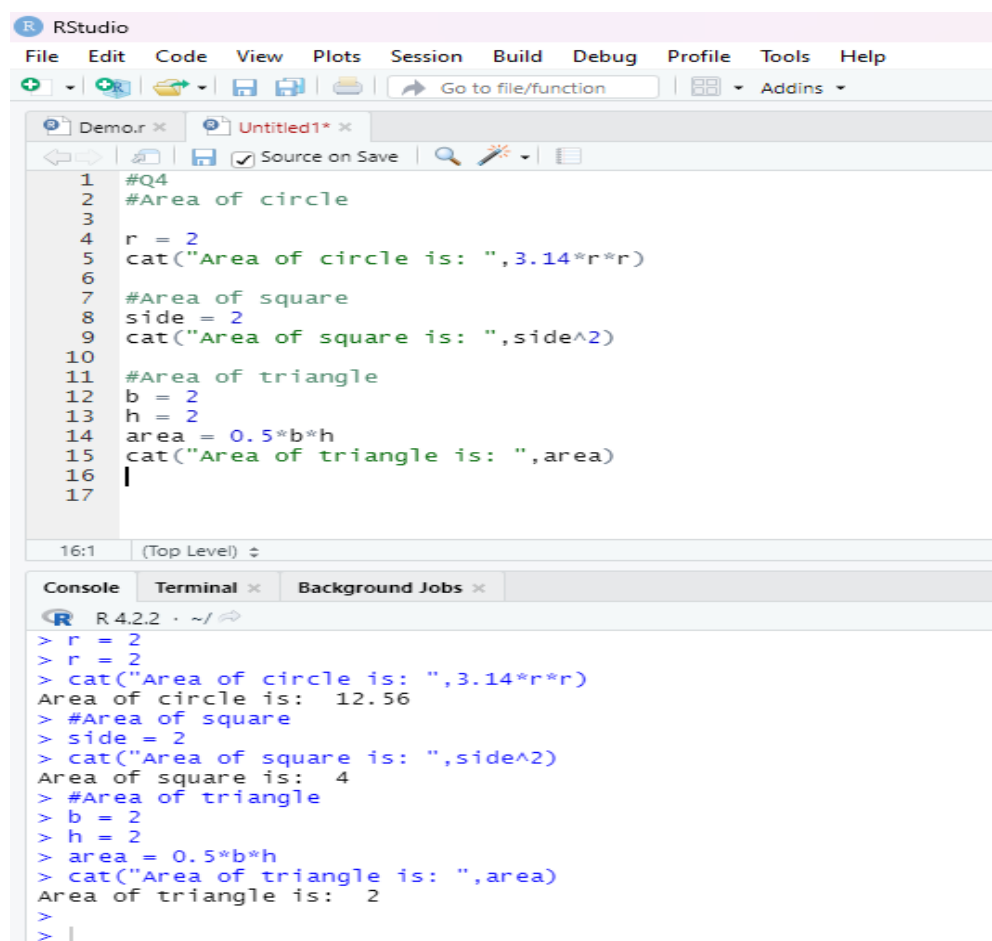
```
> cat(date())
Mon Dec 5 19:37:31 2022
>
```

3. To get the version of R on which you are working



The screenshot shows the RStudio interface. The 'Source' pane on the left contains a script with two lines: `1 print(R.version.string)` and `2` (with a cursor). The 'Console' pane on the right shows the output of the command: `> print(R.version.string)` followed by `[1] "R version 4.2.2 (2022-10-31 ucrt)"`.

4. To calculate area of a circle, square and triangle



The screenshot shows the RStudio interface with a script in the 'Source' pane and its output in the 'Console' pane. The script calculates the areas of a circle, square, and triangle with side/length 2.

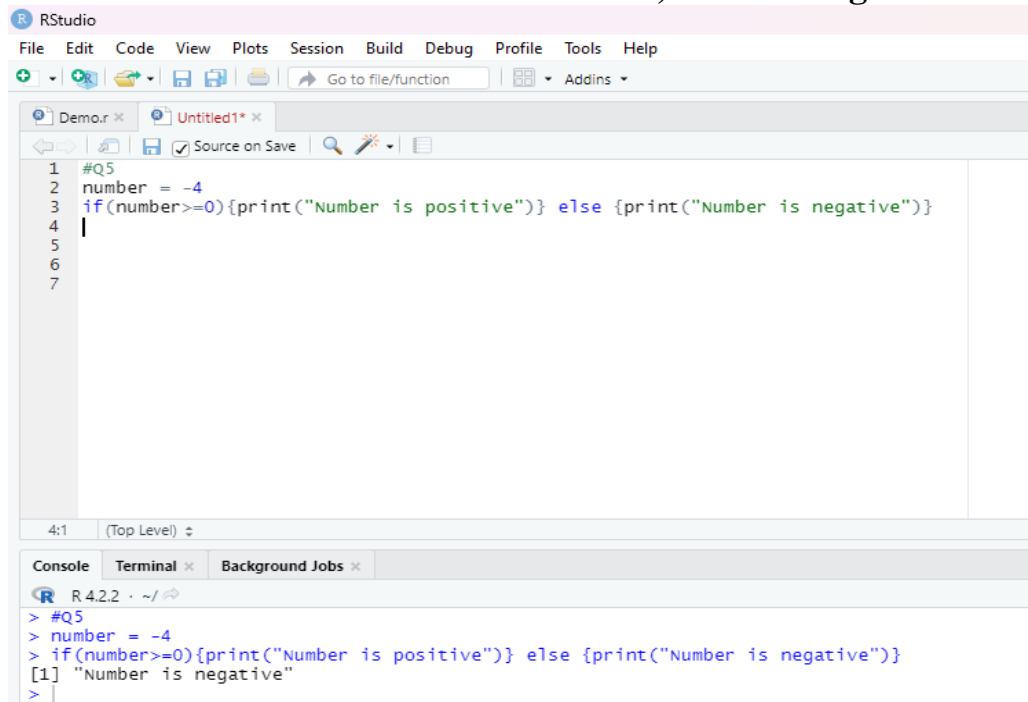
Script Code:

```
1 #Q4
2 #Area of circle
3
4 r = 2
5 cat("Area of circle is: ",3.14*r*r)
6
7 #Area of square
8 side = 2
9 cat("Area of square is: ",side^2)
10
11 #Area of triangle
12 b = 2
13 h = 2
14 area = 0.5*b*h
15 cat("Area of triangle is: ",area)
16
17
```

Console Output:

```
> r = 2
> r = 2
> cat("Area of circle is: ",3.14*r*r)
Area of circle is: 12.56
> #Area of square
> side = 2
> cat("Area of square is: ",side^2)
Area of square is: 4
> #Area of triangle
> b = 2
> h = 2
> area = 0.5*b*h
> cat("Area of triangle is: ",area)
Area of triangle is: 2
>
>
```

5. To find out whether a number is Positive, Zero or Negative.



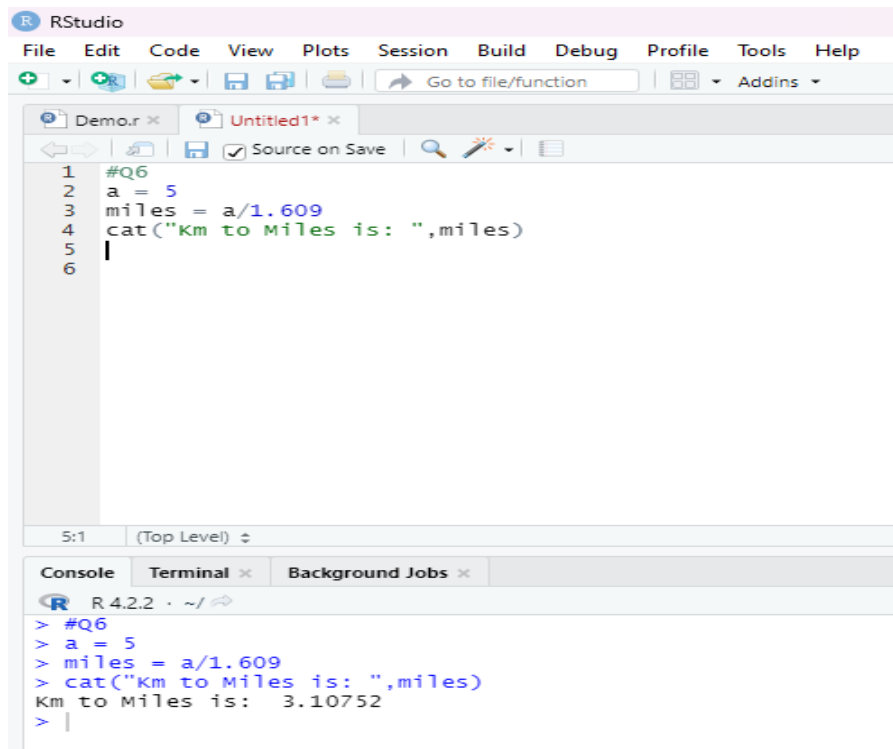
The screenshot shows the RStudio interface. The source editor contains the following R code:

```
1 #Q5
2 number = -4
3 if(number >= 0){print("Number is positive")} else {print("Number is negative")}
4 |
5
6
7
```

The console output shows the execution of the code:

```
> #Q5
> number = -4
> if(number >= 0){print("Number is positive")} else {print("Number is negative")}
[1] "Number is negative"
> |
```

6. To convert kilometers to miles



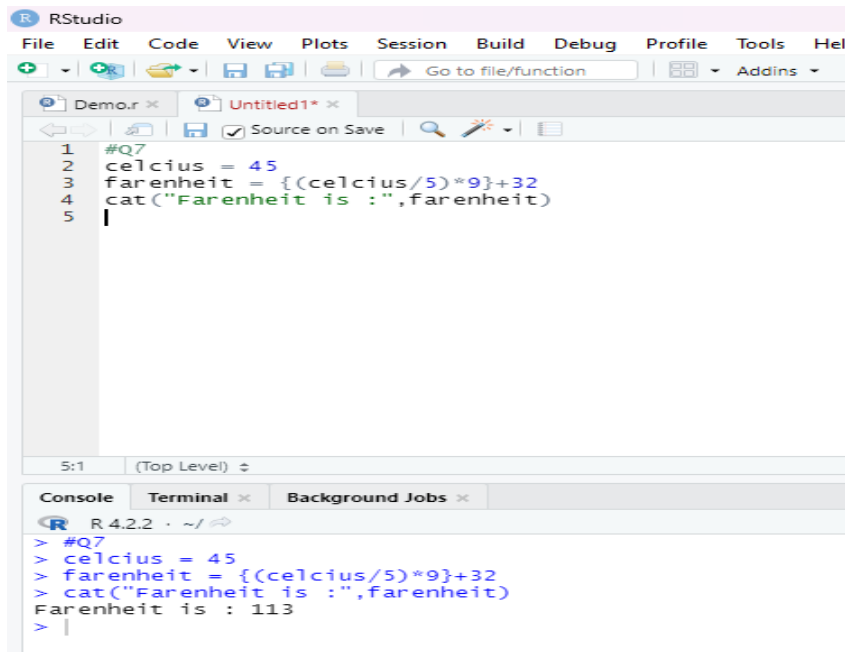
The screenshot shows the RStudio interface. The source editor contains the following R code:

```
1 #Q6
2 a = 5
3 miles = a/1.609
4 cat("km to Miles is: ",miles)
5 |
6
```

The console output shows the execution of the code:

```
> #Q6
> a = 5
> miles = a/1.609
> cat("km to Miles is: ",miles)
km to Miles is:  3.10752
> |
```

7. To convert Celsius to Fahrenheit



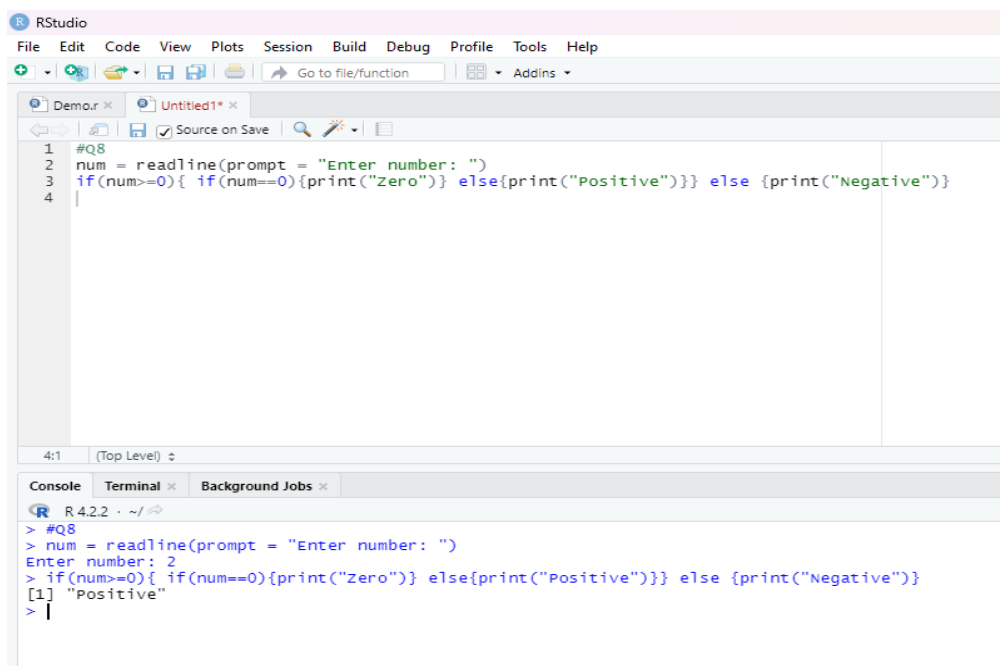
The screenshot shows the RStudio interface. The source editor contains the following R code:

```
1 #Q7
2 celcius = 45
3 fahrenheit = {(celcius/5)*9}+32
4 cat("Fahrenheit is :",fahrenheit)
5 |
```

The console shows the execution of the code:

```
> #Q7
> celcius = 45
> fahrenheit = {(celcius/5)*9}+32
> cat("Fahrenheit is :",fahrenheit)
Fahrenheit is : 113
> |
```

8. To find out whether a number is Positive, Zero or Negative. Ask user to give the input number.



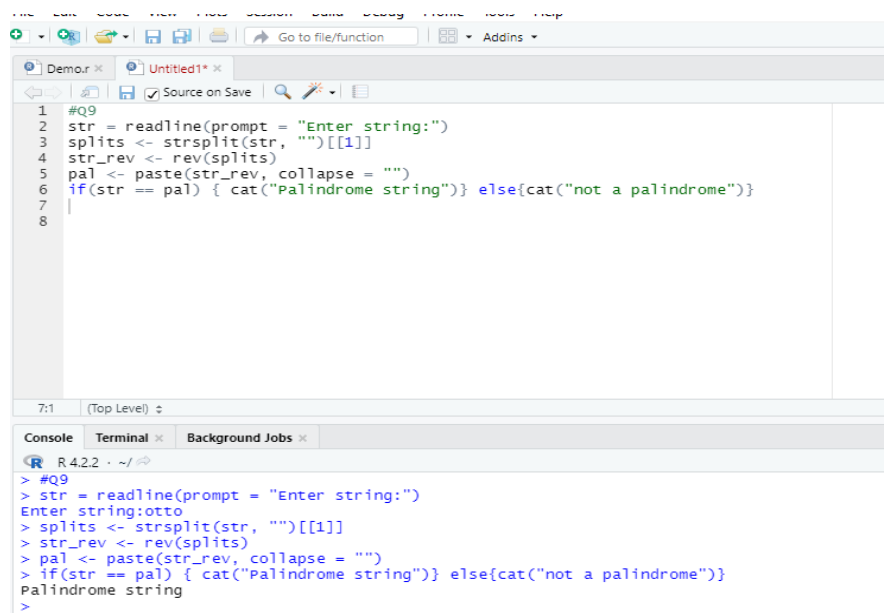
The screenshot shows the RStudio interface. The source editor contains the following R code:

```
1 #Q8
2 num = readline(prompt = "Enter number: ")
3 if(num>=0){ if(num==0){print("Zero")} else{print("Positive")}} else {print("Negative")}
4 |
```

The console shows the execution of the code:

```
> #Q8
> num = readline(prompt = "Enter number: ")
Enter number: 2
> if(num>=0){ if(num==0){print("Zero")} else{print("Positive")}} else {print("Negative")}
[1] "Positive"
> |
```

9. To find whether a string is palindrome. Ask user to give the input as string.



```

1 #Q9
2 str = readline(prompt = "Enter string:")
3 splits <- strsplit(str, "")[[1]]
4 str_rev <- rev(splits)
5 pal <- paste(str_rev, collapse = "")
6 if(str == pal) { cat("Palindrome string")} else{cat("not a palindrome")}
7
8

```

7:1 (Top Level) ↕

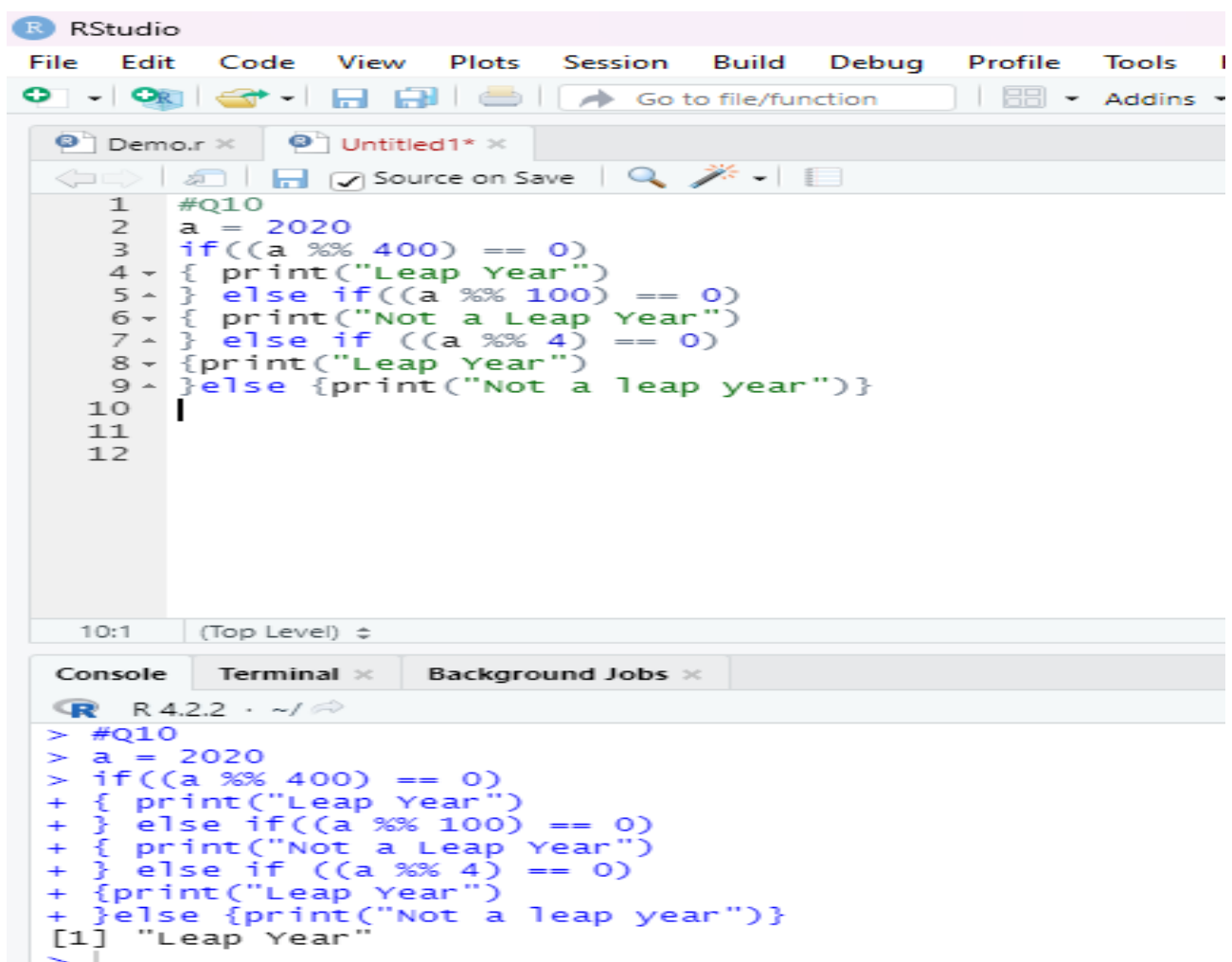
Console Terminal Background Jobs

```

R 4.2.2 ~ /
> #Q9
> str = readline(prompt = "Enter string:")
Enter string:otto
> splits <- strsplit(str, "")[[1]]
> str_rev <- rev(splits)
> pal <- paste(str_rev, collapse = "")
> if(str == pal) { cat("Palindrome string")} else{cat("not a palindrome")}
Palindrome string
>

```

10.To check if a year is leap year



```

1 #Q10
2 a = 2020
3 if((a %% 400) == 0)
4 { print("Leap Year")
5 } else if((a %% 100) == 0)
6 { print("Not a Leap Year")
7 } else if ((a %% 4) == 0)
8 {print("Leap Year")
9 }else {print("Not a leap year")}
10
11
12

```

10:1 (Top Level) ↕

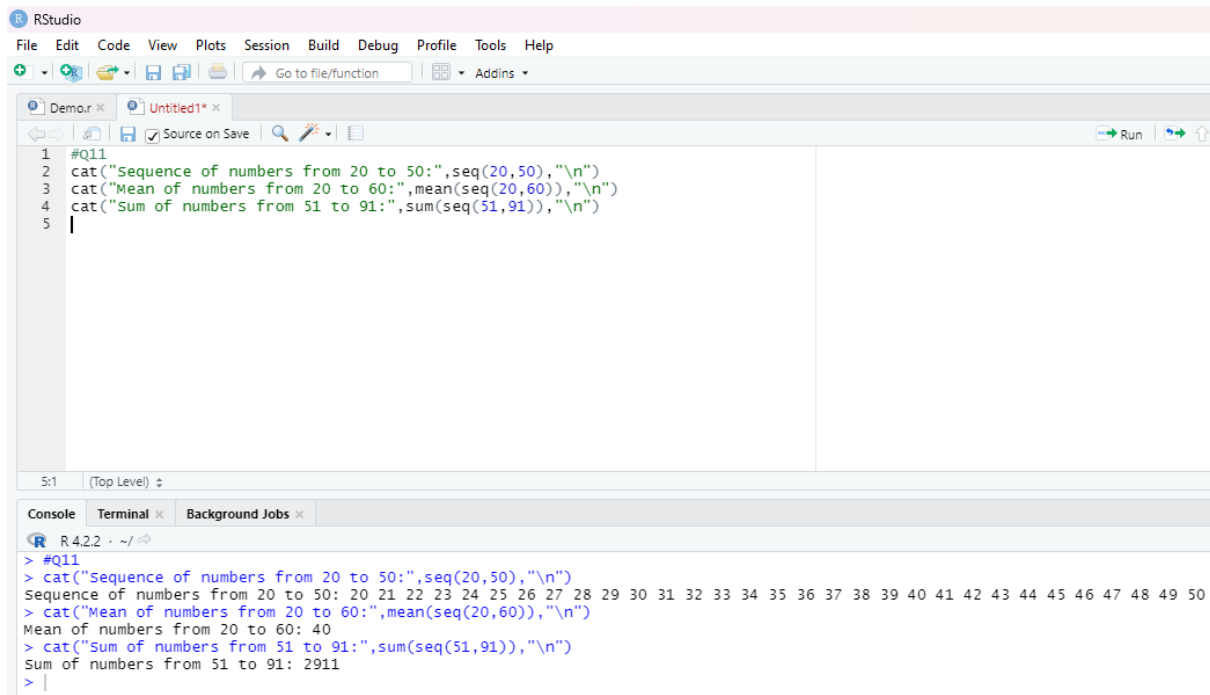
Console Terminal Background Jobs

```

R 4.2.2 ~ /
> #Q10
> a = 2020
> if((a %% 400) == 0)
+ { print("Leap Year")
+ } else if((a %% 100) == 0)
+ { print("Not a Leap Year")
+ } else if ((a %% 4) == 0)
+ {print("Leap Year")
+ }else {print("Not a leap year")}
[1] "Leap Year"
>

```

11. To create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.



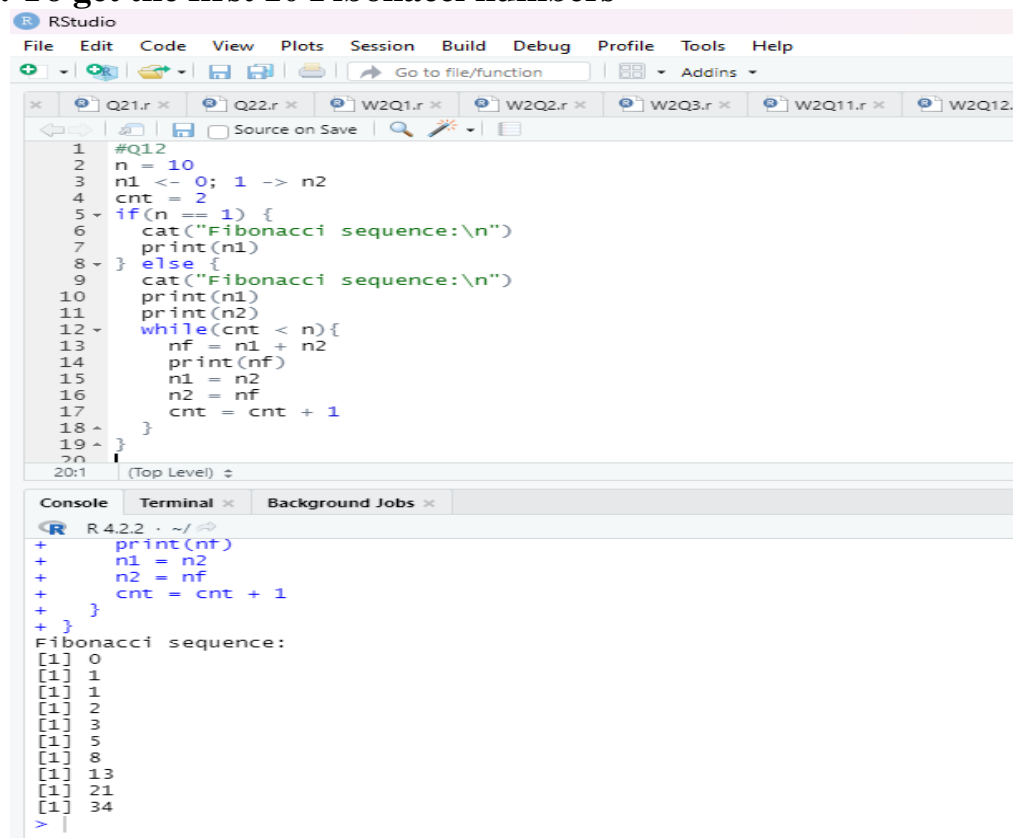
The screenshot shows the RStudio interface with a script editor and a console. The script editor contains the following R code:

```
#Q11
cat("Sequence of numbers from 20 to 50:", seq(20, 50), "\n")
cat("Mean of numbers from 20 to 60:", mean(seq(20, 60)), "\n")
cat("Sum of numbers from 51 to 91:", sum(seq(51, 91)), "\n")
```

The console shows the output of the code:

```
> #Q11
> cat("Sequence of numbers from 20 to 50:", seq(20, 50), "\n")
Sequence of numbers from 20 to 50: 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
> cat("Mean of numbers from 20 to 60:", mean(seq(20, 60)), "\n")
Mean of numbers from 20 to 60: 40
> cat("Sum of numbers from 51 to 91:", sum(seq(51, 91)), "\n")
Sum of numbers from 51 to 91: 2911
```

12. To get the first 10 Fibonacci numbers



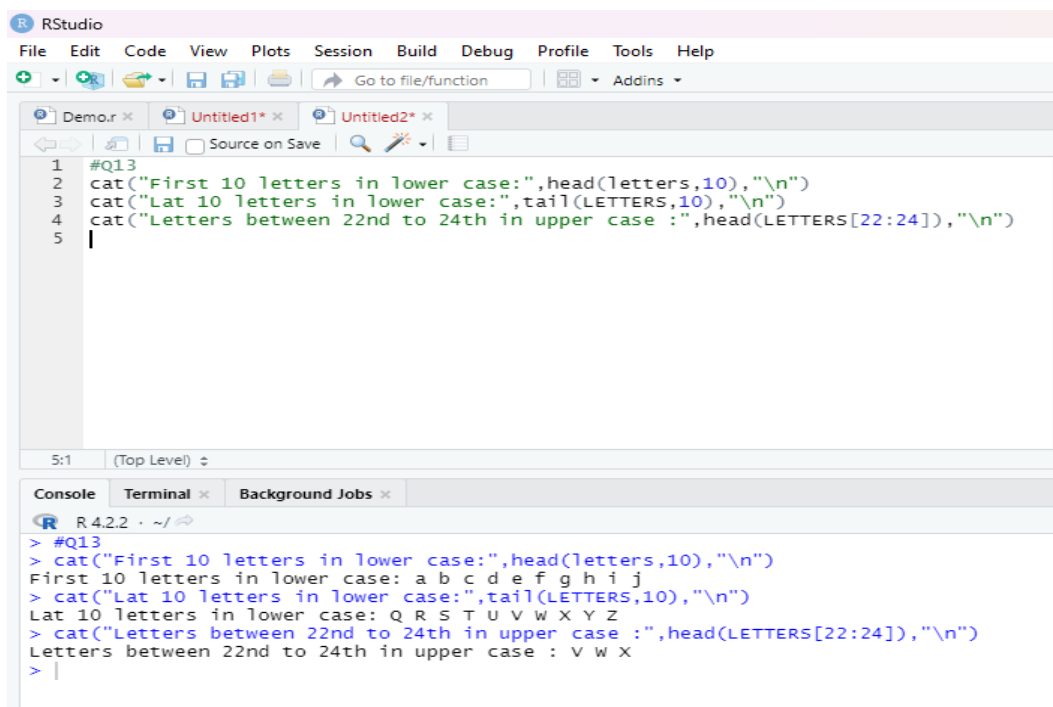
The screenshot shows the RStudio interface with a script editor and a console. The script editor contains the following R code:

```
#Q12
n = 10
n1 <- 0; n2 <- 1
cnt = 2
if(n == 1) {
  cat("Fibonacci sequence:\n")
  print(n1)
} else {
  cat("Fibonacci sequence:\n")
  print(n1)
  print(n2)
  while(cnt < n) {
    nf = n1 + n2
    print(nf)
    n1 = n2
    n2 = nf
    cnt = cnt + 1
  }
}
```

The console shows the output of the code:

```
+ print(nf)
+ n1 = n2
+ n2 = nf
+ cnt = cnt + 1
+ }
+ }
Fibonacci sequence:
[1] 0
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
[1] 13
[1] 21
[1] 34
```


13. To extract first 10 english letter in lower case and last 10 letters in uppercase and extract letters between 22nd to 24th letters in upper case



```

1 #Q13
2 cat("First 10 letters in lower case:",head(letters,10),"\n")
3 cat("Lat 10 letters in lower case:",tail(LETTERS,10),"\n")
4 cat("Letters between 22nd to 24th in upper case :",head(LETTERS[22:24]),"\n")
5 |

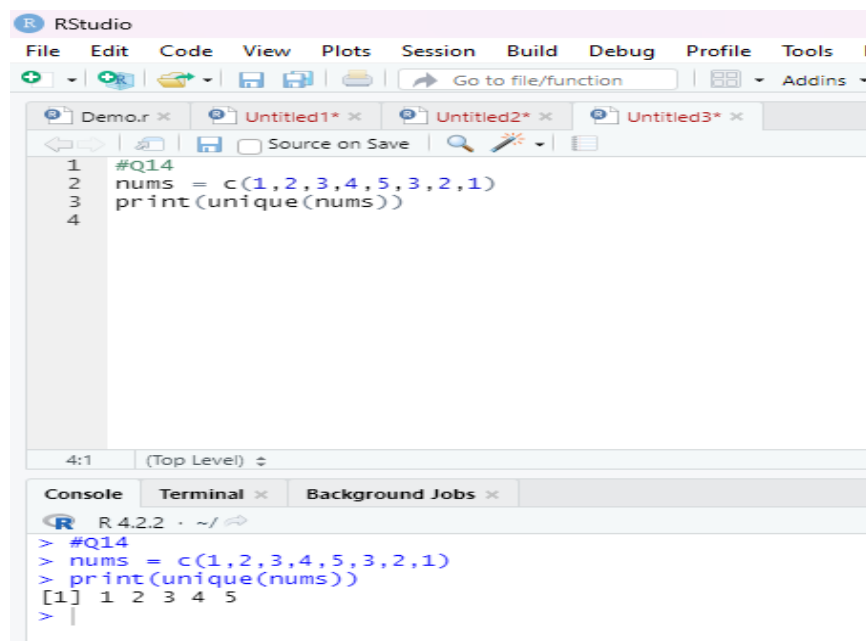
```

```

> #Q13
> cat("First 10 letters in lower case:",head(letters,10),"\n")
First 10 letters in lower case: a b c d e f g h i j
> cat("Lat 10 letters in lower case:",tail(LETTERS,10),"\n")
Lat 10 letters in lower case: Q R S T U V W X Y Z
> cat("Letters between 22nd to 24th in upper case :",head(LETTERS[22:24]),"\n")
Letters between 22nd to 24th in upper case : V W X
> |

```

14. To get the unique elements of a given string and unique numbers of vector.



```

1 #Q14
2 nums = c(1,2,3,4,5,3,2,1)
3 print(unique(nums))
4 |

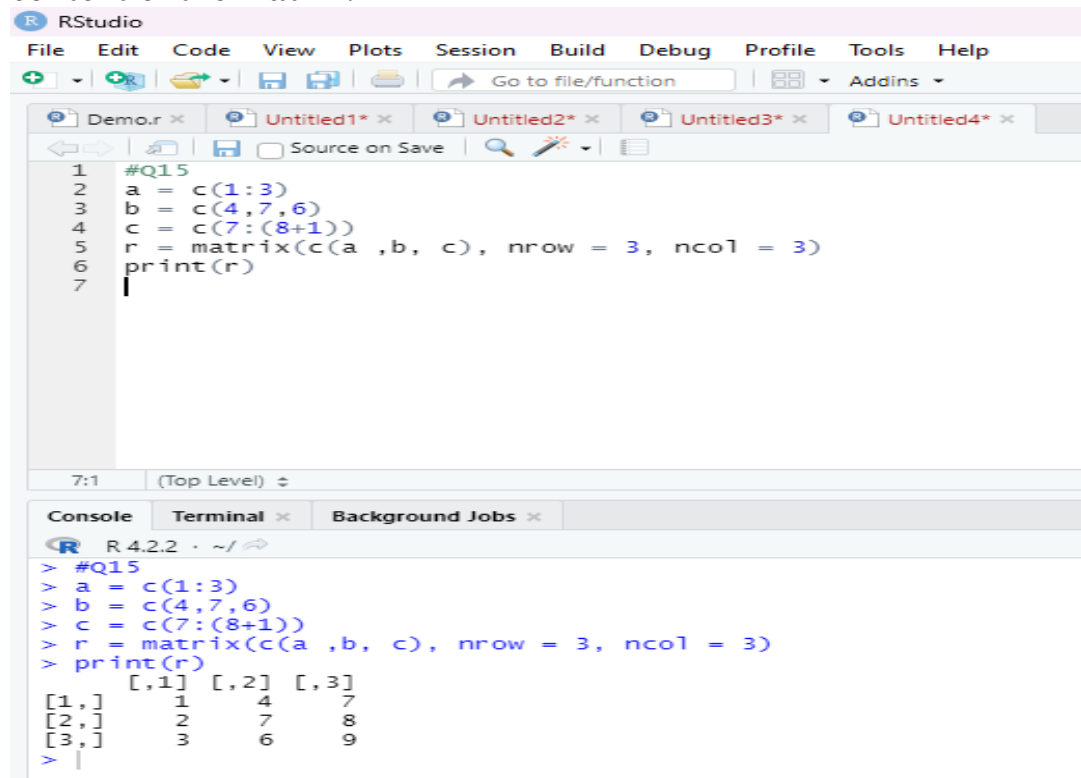
```

```

> #Q14
> nums = c(1,2,3,4,5,3,2,1)
> print(unique(nums))
[1] 1 2 3 4 5
> |

```

- 15. To create three vectors a,b,c with 3 integers.
Combine the three vectors to become a 3×3 matrix
where each column represents a vector. Print the
content of the matrix.**



```

1 #Q15
2 a = c(1:3)
3 b = c(4,7,6)
4 c = c(7:(8+1))
5 r = matrix(c(a ,b, c), nrow = 3, ncol = 3)
6 print(r)
7

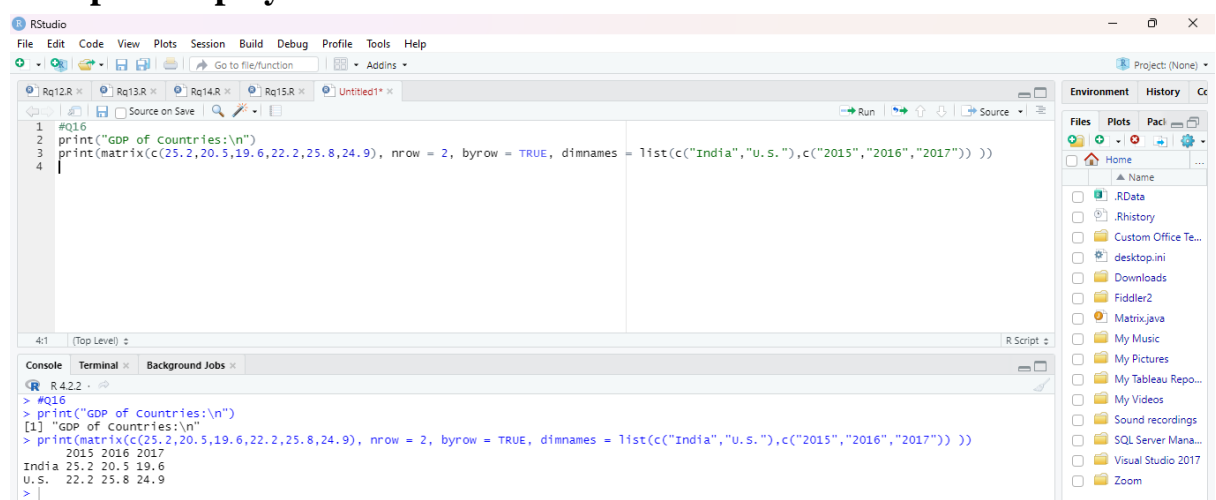
```

```

> #Q15
> a = c(1:3)
> b = c(4,7,6)
> c = c(7:(8+1))
> r = matrix(c(a ,b, c), nrow = 3, ncol = 3)
> print(r)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    7    8
[3,]    3    6    9

```

- 16. To create a matrix taking a given vector of numbers as
input. Display the matrix.**



```

1 #Q16
2 print("GDP of Countries:\n")
3 print(matrix(c(25.2,20.5,19.6,22.2,25.8,24.9), nrow = 2, byrow = TRUE, dimnames = list(c("India","U.S."),c("2015","2016","2017")) ))
4

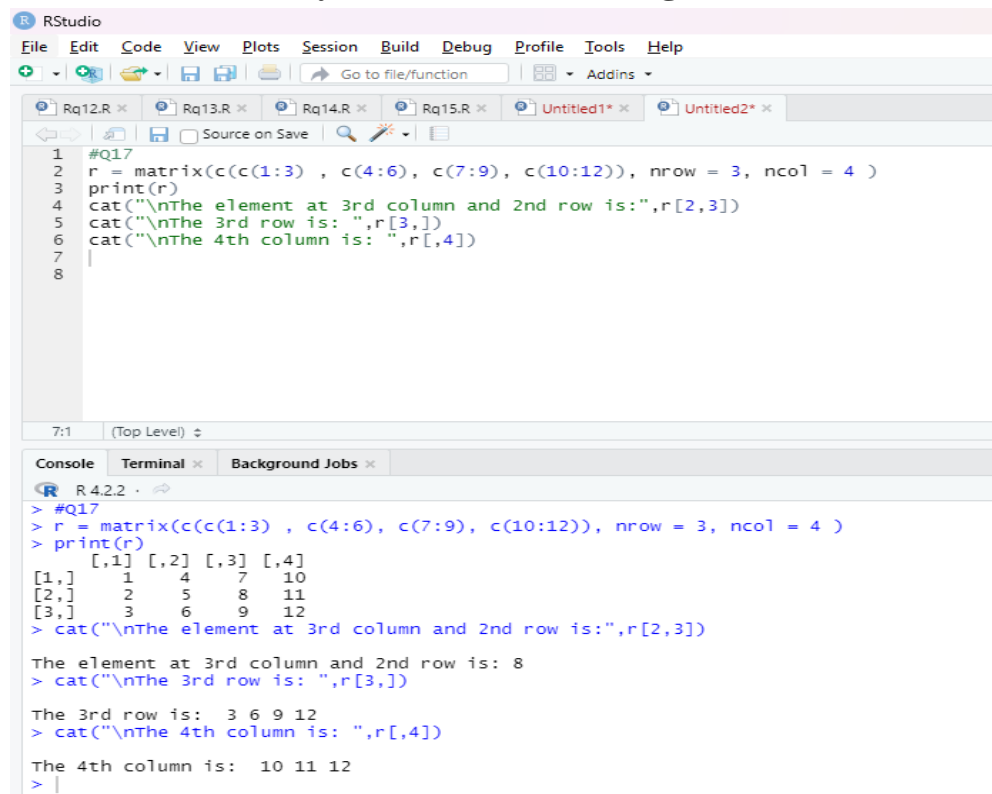
```

```

> #Q16
> print("GDP of Countries:\n")
[1] "GDP of Countries:\n"
> print(matrix(c(25.2,20.5,19.6,22.2,25.8,24.9), nrow = 2, byrow = TRUE, dimnames = list(c("India","U.S."),c("2015","2016","2017")) ))
      2015 2016 2017
India 25.2 20.5 19.6
U.S.  22.2 25.8 24.9

```

17. To access the element at 3rd column and 2nd row, only the 3rd row and only the 4th column of a given matrix.



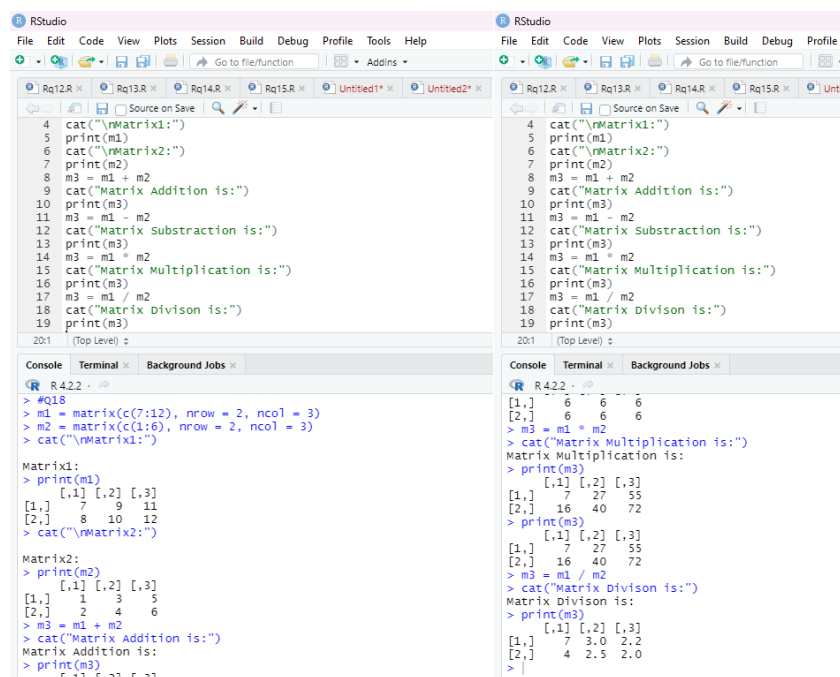
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Source on Save Go to file/function Addins
Rq12.R Rq13.R Rq14.R Rq15.R Untitled1* Untitled2*
1 #Q17
2 r = matrix(c(c(1:3), c(4:6), c(7:9), c(10:12)), nrow = 3, ncol = 4)
3 print(r)
4 cat("\nThe element at 3rd column and 2nd row is:", r[2,3])
5 cat("\nThe 3rd row is: ", r[3,])
6 cat("\nThe 4th column is: ", r[,4])
7
8

7:1 (Top Level)
Console Terminal Background Jobs
R 4.2.2
> #Q17
> r = matrix(c(c(1:3), c(4:6), c(7:9), c(10:12)), nrow = 3, ncol = 4)
> print(r)
      [,1] [,2] [,3] [,4]
[1,]  1    4    7   10
[2,]  2    5    8   11
[3,]  3    6    9   12
> cat("\nThe element at 3rd column and 2nd row is:", r[2,3])
The element at 3rd column and 2nd row is: 8
> cat("\nThe 3rd row is: ", r[3,])
The 3rd row is:  3 6 9 12
> cat("\nThe 4th column is: ", r[,4])
The 4th column is:  10 11 12
>

```

18. To create two 2x3 matrix and add, subtract, multiply and divide thematrixes

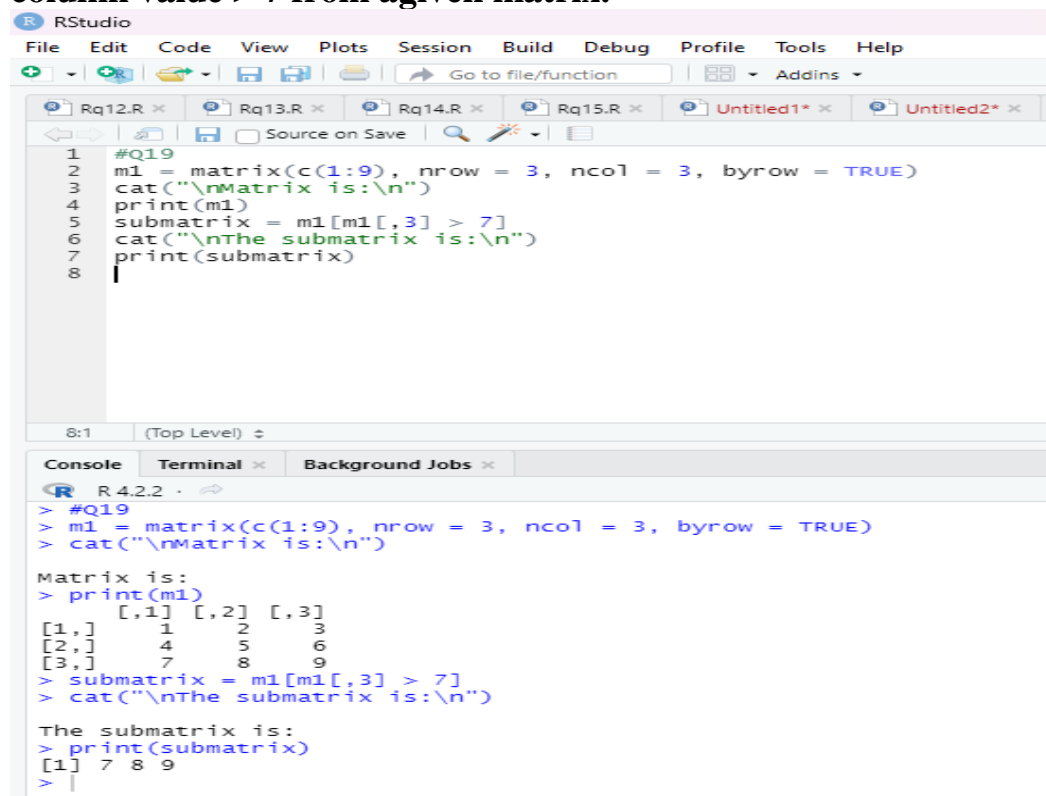


```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Source on Save Go to file/function Addins
Rq12.R Rq13.R Rq14.R Rq15.R Untitled1* Untitled2*
4 cat("\nMatrix1:")
5 print(m1)
6 cat("\nMatrix2:")
7 print(m2)
8 m3 = m1 + m2
9 cat("Matrix Addition is:")
10 print(m3)
11 m3 = m1 - m2
12 cat("Matrix Substraction is:")
13 print(m3)
14 m3 = m1 * m2
15 cat("Matrix Multiplication is:")
16 print(m3)
17 m3 = m1 / m2
18 cat("Matrix Divison is:")
19 print(m3)
20:1 (Top Level)
Console Terminal Background Jobs
R 4.2.2
> #Q18
> m1 = matrix(c(7:12), nrow = 2, ncol = 3)
> m2 = matrix(c(1:6), nrow = 2, ncol = 3)
> cat("\nMatrix1:")
Matrix1:
> print(m1)
      [,1] [,2] [,3]
[1,]  7    9   11
[2,]  8   10   12
> cat("\nMatrix2:")
Matrix2:
> print(m2)
      [,1] [,2] [,3]
[1,]  1    3    5
[2,]  2    4    6
> m3 = m1 + m2
> cat("Matrix Addition is:")
Matrix Addition is:
> print(m3)
      [,1] [,2] [,3]
[1,]  8   12   16
[2,]  9   14   18
> m3 = m1 - m2
> cat("Matrix Substraction is:")
Matrix Substraction is:
> print(m3)
      [,1] [,2] [,3]
[1,]  5    6    6
[2,]  6    6    6
> m3 = m1 * m2
> cat("Matrix Multiplication is:")
Matrix Multiplication is:
> print(m3)
      [,1] [,2] [,3]
[1,]  7   27  55
[2,] 16   40  72
> m3 = m1 / m2
> cat("Matrix Division is:")
Matrix Division is:
> print(m3)
      [,1] [,2] [,3]
[1,]  7   3.0  2.2
[2,]  4   2.5  2.0
>

```

19. To extract the submatrix whose rows have column value > 7 from agiven matrix.



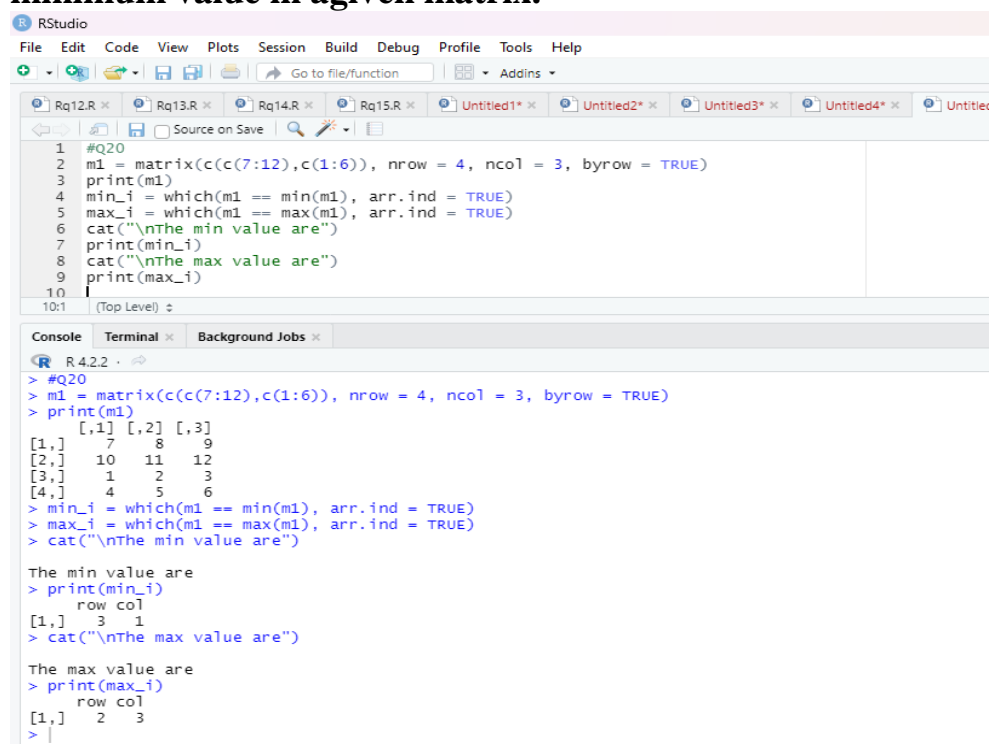
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Rq12.R x Rq13.R x Rq14.R x Rq15.R x Untitled1* x Untitled2* x
Source on Save
1 #Q19
2 m1 = matrix(c(1:9), nrow = 3, ncol = 3, byrow = TRUE)
3 cat("\nMatrix is:\n")
4 print(m1)
5 submatrix = m1[m1[,3] > 7]
6 cat("\nThe submatrix is:\n")
7 print(submatrix)
8

Console Terminal x Background Jobs x
R 4.2.2
> #Q19
> m1 = matrix(c(1:9), nrow = 3, ncol = 3, byrow = TRUE)
> cat("\nMatrix is:\n")
Matrix is:
> print(m1)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> submatrix = m1[m1[,3] > 7]
> cat("\nThe submatrix is:\n")
The submatrix is:
> print(submatrix)
[1] 7 8 9
>

```

20. To find row and column index of maximum and minimum value in agiven matrix.



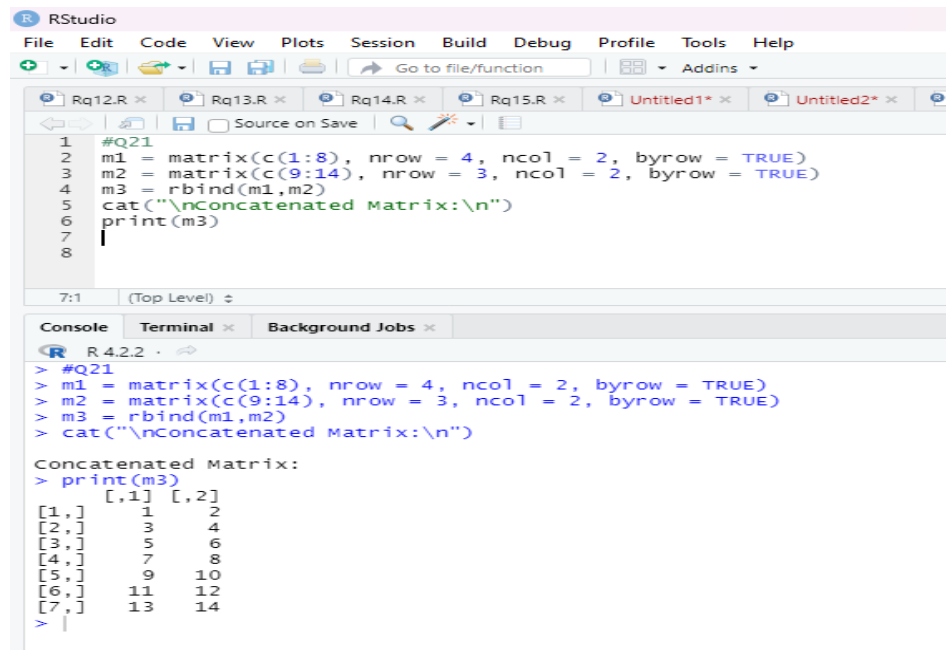
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Rq12.R x Rq13.R x Rq14.R x Rq15.R x Untitled1* x Untitled2* x Untitled3* x Untitled4* x Untitled5* x
Source on Save
1 #Q20
2 m1 = matrix(c(c(7:12),c(1:6)), nrow = 4, ncol = 3, byrow = TRUE)
3 print(m1)
4 min_i = which(m1 == min(m1), arr.ind = TRUE)
5 max_i = which(m1 == max(m1), arr.ind = TRUE)
6 cat("\nThe min value are")
7 print(min_i)
8 cat("\nThe max value are")
9 print(max_i)
10
10:1 (Top Level)

Console Terminal x Background Jobs x
R 4.2.2
> #Q20
> m1 = matrix(c(c(7:12),c(1:6)), nrow = 4, ncol = 3, byrow = TRUE)
> print(m1)
      [,1] [,2] [,3]
[1,]    7    8    9
[2,]   10   11   12
[3,]    1    2    3
[4,]    4    5    6
> min_i = which(m1 == min(m1), arr.ind = TRUE)
> max_i = which(m1 == max(m1), arr.ind = TRUE)
> cat("\nThe min value are")
The min value are
> print(min_i)
      row col
[1,]    3    1
> cat("\nThe max value are")
The max value are
> print(max_i)
      row col
[1,]    2    3
>

```

21. To concatenate two given matrices of same column but different rows



The screenshot shows the RStudio interface. The script editor contains the following R code:

```

1 #Q21
2 m1 = matrix(c(1:8), nrow = 4, ncol = 2, byrow = TRUE)
3 m2 = matrix(c(9:14), nrow = 3, ncol = 2, byrow = TRUE)
4 m3 = rbind(m1,m2)
5 cat("\nConcatenated Matrix:\n")
6 print(m3)
7
8

```

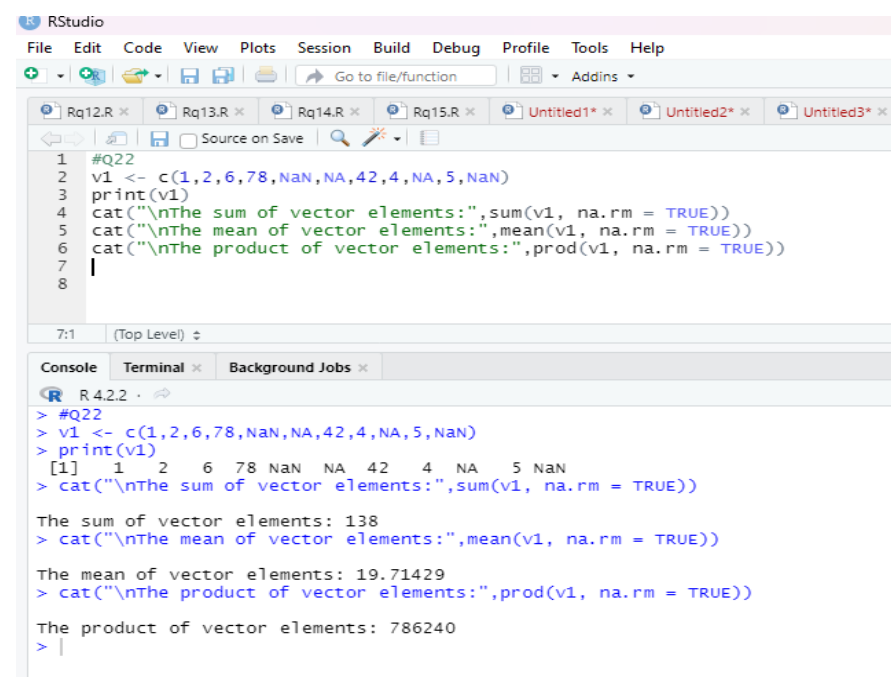
The console output shows the execution of the code:

```

> #Q21
> m1 = matrix(c(1:8), nrow = 4, ncol = 2, byrow = TRUE)
> m2 = matrix(c(9:14), nrow = 3, ncol = 2, byrow = TRUE)
> m3 = rbind(m1,m2)
> cat("\nConcatenated Matrix:\n")
Concatenated Matrix:
> print(m3)
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
[6,]   11   12
[7,]   13   14

```

22. To find Sum, Mean and Product of a Vector, ignore element like NA or NaN.



The screenshot shows the RStudio interface. The script editor contains the following R code:

```

1 #Q22
2 v1 <- c(1,2,6,78,NaN,NA,42,4,NA,5,NaN)
3 print(v1)
4 cat("\nThe sum of vector elements:",sum(v1, na.rm = TRUE))
5 cat("\nThe mean of vector elements:",mean(v1, na.rm = TRUE))
6 cat("\nThe product of vector elements:",prod(v1, na.rm = TRUE))
7
8

```

The console output shows the execution of the code:

```

> #Q22
> v1 <- c(1,2,6,78,NaN,NA,42,4,NA,5,NaN)
> print(v1)
[1] 1 2 6 78 NaN NA 42 4 NA 5 NaN
> cat("\nThe sum of vector elements:",sum(v1, na.rm = TRUE))
The sum of vector elements: 138
> cat("\nThe mean of vector elements:",mean(v1, na.rm = TRUE))
The mean of vector elements: 19.71429
> cat("\nThe product of vector elements:",prod(v1, na.rm = TRUE))
The product of vector elements: 786240

```

23. To sort a Vector in ascending and descending order.

The screenshot displays the RStudio interface. The top pane shows an R script with the following code:

```

1 #Q23
2 v1 <- c(-1,2.5,6.47,NaN,NA,42,4.22,5)
3 print(v1)
4 cat("\nThe vector in ascending order:\n")
5 print(sort(v1))
6 cat("\nThe vector in descending order:\n")
7 print(sort(v1,decreasing = TRUE))
8
9

```

The bottom pane shows the console output of the script:

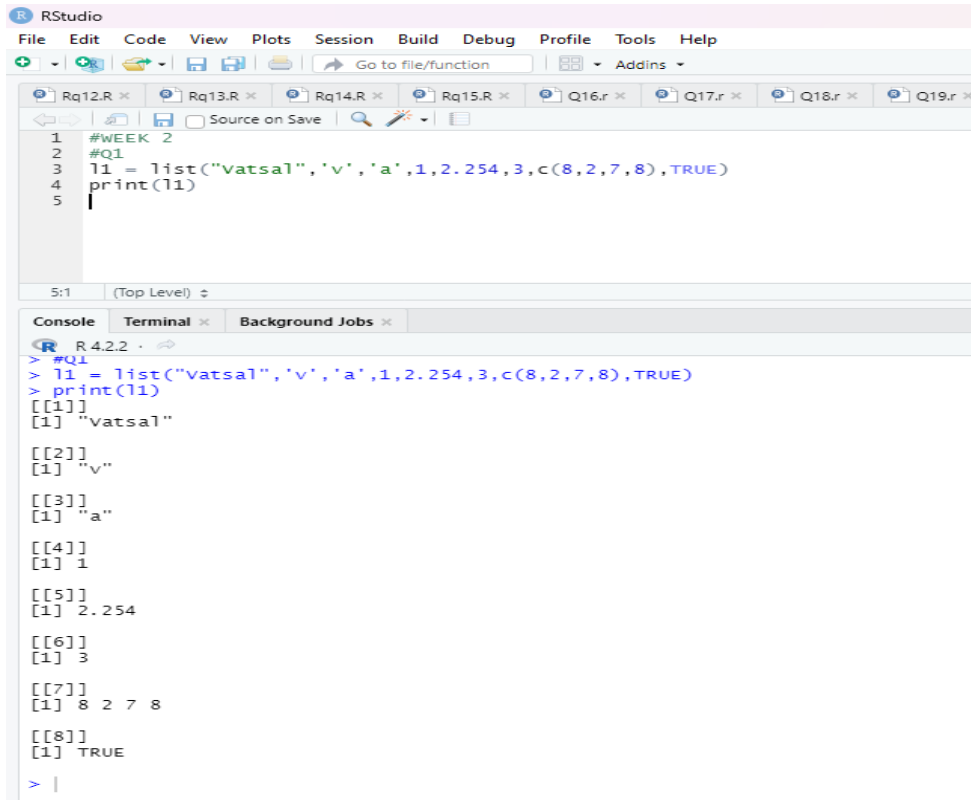
```

> #Q23
> v1 <- c(-1,2.5,6.47,NaN,NA,42,4.22,5)
> print(v1)
[1] -1.00 2.50 6.47 NaN NA 42.00 4.22 5.00
> cat("\nThe vector in ascending order:\n")
The vector in ascending order:
> print(sort(v1))
[1] -1.00 2.50 4.22 5.00 6.47 42.00
> cat("\nThe vector in descending order:\n")
The vector in descending order:
> print(sort(v1,decreasing = TRUE))
[1] 42.00 6.47 5.00 4.22 2.50 -1.00
>

```

Week 2

1. To create a list containing strings, numbers, vectors and a logical values



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Rq12.R x Rq13.R x Rq14.R x Rq15.R x Q16.r x Q17.r x Q18.r x Q19.r x
1 #WEEK 2
2 #Q1
3 l1 = list("Vatsal", 'v', 'a', 1, 2.254, 3, c(8, 2, 7, 8), TRUE)
4 print(l1)
5

5:1 (Top Level)
Console Terminal Background Jobs
R 4.2.2
> #Q1
> l1 = list("Vatsal", 'v', 'a', 1, 2.254, 3, c(8, 2, 7, 8), TRUE)
> print(l1)
[[1]]
[1] "Vatsal"

[[2]]
[1] "v"

[[3]]
[1] "a"

[[4]]
[1] 1

[[5]]
[1] 2.254

[[6]]
[1] 3

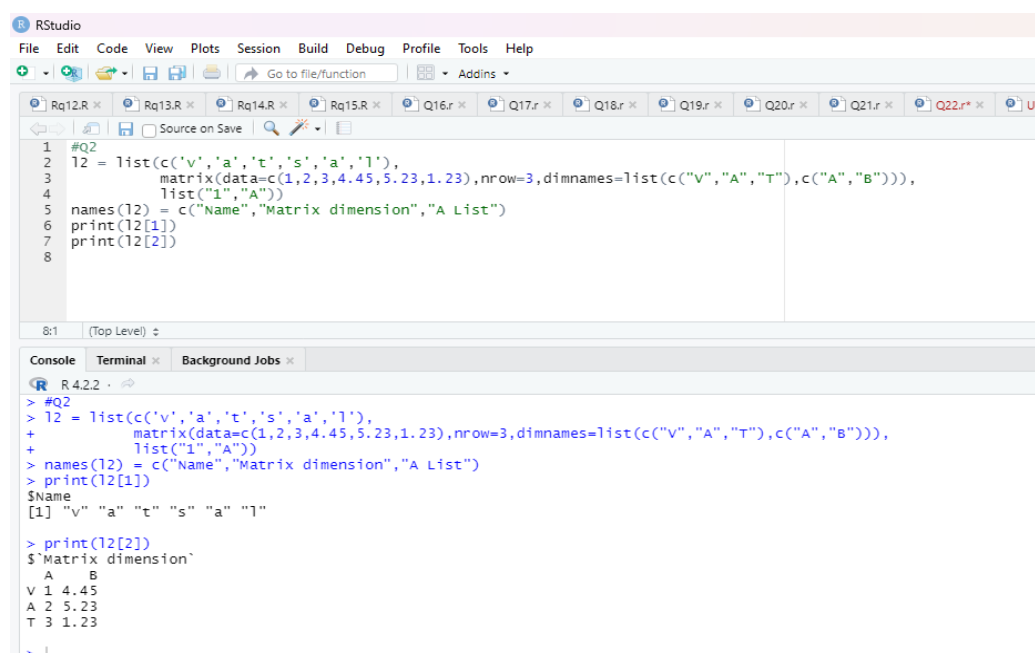
[[7]]
[1] 8 2 7 8

[[8]]
[1] TRUE

>

```

2. To list containing a vector, a matrix and a list and give names to the elements in thelist. Access the first and second element of the list.



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Rq12.R x Rq13.R x Rq14.R x Rq15.R x Q16.r x Q17.r x Q18.r x Q19.r x Q20.r x Q21.r x Q22.r* x Ur
1 #Q2
2 l2 = list(c('v', 'a', 't', 's', 'a', 'l'),
3           matrix(data=c(1,2,3,4,45,5,23,1,23), nrow=3, dimnames=list(c("v", "A", "T"), c("A", "B"))),
4           list("1", "A"))
5 names(l2) = c("Name", "Matrix dimension", "A List")
6 print(l2[1])
7 print(l2[2])
8

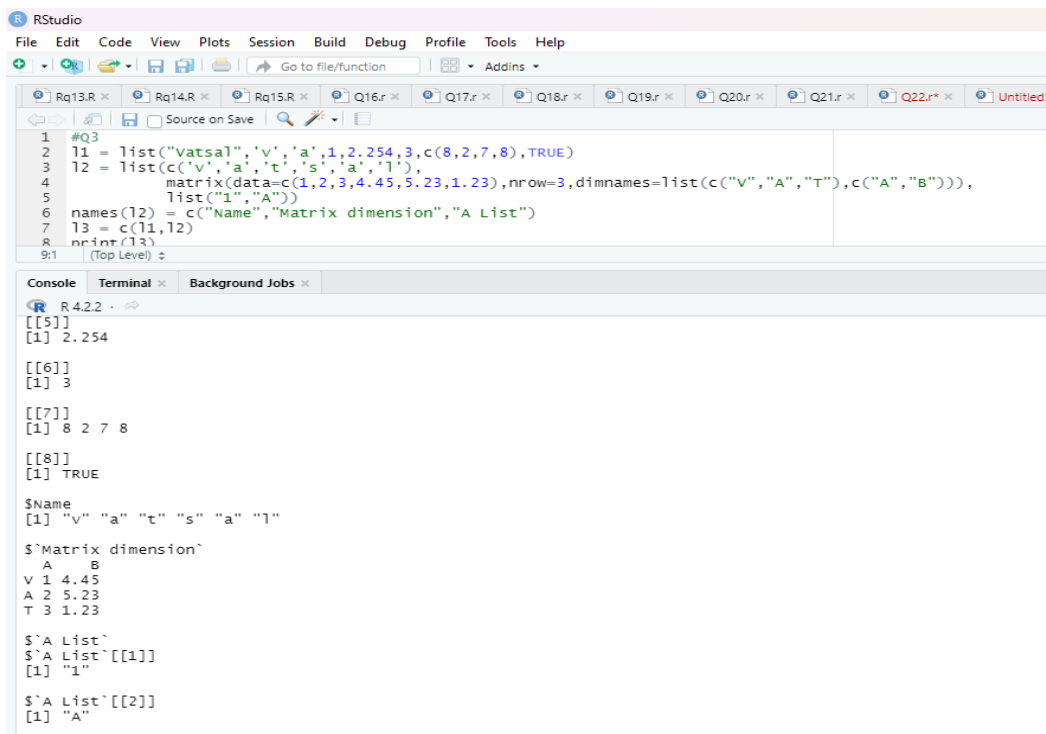
8:1 (Top Level)
Console Terminal Background Jobs
R 4.2.2
> #Q2
> l2 = list(c('v', 'a', 't', 's', 'a', 'l'),
+           matrix(data=c(1,2,3,4,45,5,23,1,23), nrow=3, dimnames=list(c("v", "A", "T"), c("A", "B"))),
+           list("1", "A"))
> names(l2) = c("Name", "Matrix dimension", "A List")
> print(l2[1])
$name
[1] "v" "a" "t" "s" "a" "l"

> print(l2[2])
$`Matrix dimension`
  A  B
v 1  4.45
A 2  5.23
T 3  1.23

>

```

3. To merge two given lists into one list.



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Rq13.R Rq14.R Rq15.R Rq16.R Rq17.R Rq18.R Rq19.R Q20.R Q21.R Q22.R* Untitled
1 #Q3
2 l1 = list("vatsal", 'v', 'a', 1, 2.254, 3, c(8, 2, 7, 8), TRUE)
3 l2 = list(c('v', 'a', 't', 's', 'a', 'l'),
4         matrix(data=c(1, 2, 3, 4.45, 5.23, 1.23), nrow=3, dimnames=list(c("v", "A", "T"), c("A", "B"))),
5         list("1", "A"))
6 names(l2) = c("Name", "Matrix dimension", "A List")
7 l3 = c(l1, l2)
8 print(l3)
9:1 (Top Level) ↕

Console Terminal Background Jobs
R 4.2.2
[[5]]
[1] 2.254

[[6]]
[1] 3

[[7]]
[1] 8 2 7 8

[[8]]
[1] TRUE

$Name
[1] "v" "a" "t" "s" "a" "l"

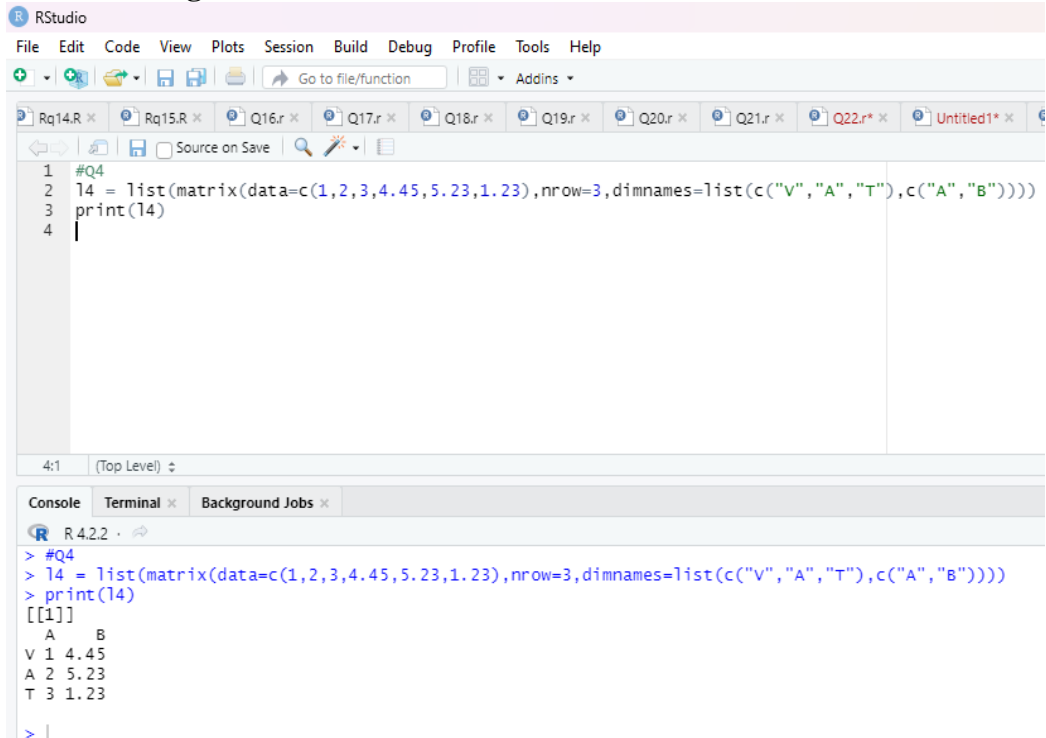
$`Matrix dimension`
  A B
v 1 4.45
A 2 5.23
T 3 1.23

$`A List`
$`A List`[[1]]
[1] "1"

$`A List`[[2]]
[1] "A"

```

4. To convert a given matrix to a list.



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Rq14.R Rq15.R Rq16.R Rq17.R Rq18.R Rq19.R Q20.R Q21.R Q22.R* Untitled1*
1 #Q4
2 l4 = list(matrix(data=c(1, 2, 3, 4.45, 5.23, 1.23), nrow=3, dimnames=list(c("v", "A", "T"), c("A", "B"))))
3 print(l4)
4 |

4:1 (Top Level) ↕

Console Terminal Background Jobs
R 4.2.2
> #Q4
> l4 = list(matrix(data=c(1, 2, 3, 4.45, 5.23, 1.23), nrow=3, dimnames=list(c("v", "A", "T"), c("A", "B"))))
> print(l4)
[[1]]
  A B
v 1 4.45
A 2 5.23
T 3 1.23

> |

```


5. To assign NULL to a given list element.

The screenshot shows the RStudio interface. The source editor contains the following R code:

```

1 #Q5
2 l5 = list(1,2,"ABC","h","v12")
3 l5[4] <- list(NULL)
4 print(l5)
5
6

```

The console output shows the result of the code execution:

```

> #Q5
> l5 = list(1,2,"ABC","h","v12")
> l5[4] <- list(NULL)
> print(l5)
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] "ABC"

[[4]]
NULL

[[5]]
[1] "v12"

>

```

6. To create an empty data frame.

The screenshot shows the RStudio interface. The source editor contains the following R code:

```

1 #Q6
2 df1 = data.frame(Name=character(),PRN=numeric(),course=character())
3 print(df1)
4

```

The console output shows the result of the code execution:

```

> #Q6
> df1 = data.frame(Name=character(),PRN=numeric(),course=character())
> print(df1)
[1] Name    PRN      course
<0 rows> (or 0-length row.names)
>

```

7. To create a data frame from four given vectors.

The screenshot shows the RStudio interface. The script editor contains the following code:

```
#Q7
df2 = data.frame(Name=c("A", "B", "C"), PRN=c(11322, 11321, 11320), Course=c("MSc(IT)", "BCA", "BSc(IT)"))
print(df2)
```

The console output shows the resulting data frame:

```
> #Q7
> df2 = data.frame(Name=c("A", "B", "C"), PRN=c(11322, 11321, 11320), Course=c("MSc(IT)", "BCA", "BSc(IT)"))
> print(df2)
  Name PRN Course
1   A 11322 MSc(IT)
2   B 11321  BCA
3   C 11320 BSc(IT)
```

8. To get the statistical summary and nature of the data of a given data frame.

The screenshot shows the RStudio interface. The script editor contains the following code:

```
#Q8
setwd("C:/Users/DELL/Downloads")
df1 = head(read.csv("winequality-red.csv"), 50)
print(summary(df1))
```

The console output shows the statistical summary of the data frame:

```
> #Q8
> setwd("C:/Users/DELL/Downloads")
> df1 = head(read.csv("winequality-red.csv"), 50)
> print(summary(df1))
      ID      fixed.acidity      volatile.acidity
Min.   : 1.00      Min.   : 4.60      Min.   : 0.2200
1st Qu.:13.25      1st Qu.: 6.95      1st Qu.: 0.4150
Median :25.50      Median : 7.60      Median : 0.5850
Mean   :25.50      Mean   : 7.51      Mean   : 0.5520
3rd Qu.:37.75      3rd Qu.: 7.90      3rd Qu.: 0.6538
Max.   :50.00      Max.   :11.20      Max.   :1.1300

citric.acid      residual.sugar      chlorides
Min.   :0.0000      Min.   : 1.200      Min.   :0.0500
1st Qu.:0.0450      1st Qu.: 1.800      1st Qu.:0.0740
Median :0.1700      Median : 2.000      Median :0.0830
Mean   :0.1932      Mean   : 2.644      Mean   :0.1050
3rd Qu.:0.2975      3rd Qu.: 2.475      3rd Qu.:0.1005
Max.   :0.5600      Max.   :10.700      Max.   :0.3680
NA's    :3

free.sulfur.dioxide      total.sulfur.dioxide      density
Min.   : 3.00      Min.   : 11.00      Min.   :0.9934
1st Qu.: 9.00      1st Qu.: 29.00      1st Qu.:0.9964
Median :13.00      Median : 48.00      Median :0.9968
Mean   :15.69      Mean   : 54.34      Mean   :0.9968
3rd Qu.:17.00      3rd Qu.: 70.00      3rd Qu.:0.9978
Max.   :52.00      Max.   :148.00      Max.   :0.9993
NA's    :2

pH      sulphates      alcohol
Min.   :3.040      Min.   :0.4600      Min.   : 9.000
1st Qu.:3.250      1st Qu.:0.5500      1st Qu.: 9.325
Median :3.335      Median :0.5800      Median : 9.500
Mean   :3.337      Mean   :0.6836      Mean   : 9.742
3rd Qu.:3.415      3rd Qu.:0.7875      3rd Qu.: 9.875
```

9. To extract specific column from a data frame using column name

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Rq14.R Rq15.R Rq16.R Rq17.R Rq18.R Rq19.R Rq20.R Rq21.R Rq22.R
Source on Save
1 #Q8
2 setwd("C:/Users/DELL/Downloads")
3 df1 = head(read.csv("winequality-red.csv"),50)
4 print(summary(df1))
5
5:1 (Top Level)
Console Terminal Background Jobs
R 4.2.2 C:/Users/DELL/Downloads/
> #Q8
> setwd("C:/Users/DELL/Downloads")
> df1 = head(read.csv("winequality-red.csv"),50)
> print(summary(df1))
      ID      fixed.acidity    volatile.acidity
Min.   : 1.00      Min.   : 4.60      Min.   :0.2200
1st Qu.:13.25      1st Qu.: 6.95      1st Qu.:0.4150
Median :25.50      Median : 7.60      Median :0.5850
Mean   :25.50      Mean   : 7.51      Mean   :0.5520
3rd Qu.:37.75      3rd Qu.: 7.90      3rd Qu.:0.6538
Max.   :50.00      Max.   :11.20      Max.   :1.1300

      citric.acid    residual.sugar    chlorides
Min.   :0.0000      Min.   : 1.200      Min.   :0.0500
1st Qu.:0.0450      1st Qu.: 1.800      1st Qu.:0.0740
Median :0.1700      Median : 2.000      Median :0.0830
Mean   :0.1932      Mean   : 2.644      Mean   :0.1050
3rd Qu.:0.2975      3rd Qu.: 2.475      3rd Qu.:0.1005
Max.   :0.5600      Max.   :10.700      Max.   :0.3680
NA's   :3

      free.sulfur.dioxide    total.sulfur.dioxide    density
Min.   : 3.00      Min.   : 11.00      Min.   :0.9934
1st Qu.: 9.00      1st Qu.: 29.00      1st Qu.:0.9964
Median :13.00      Median : 48.00      Median :0.9968
Mean   :15.69      Mean   : 54.34      Mean   :0.9968
3rd Qu.:17.00      3rd Qu.: 70.00      3rd Qu.:0.9978
Max.   :52.00      Max.   :148.00      Max.   :0.9993
NA's   :2

      pH      sulphates    alcohol
Min.   :3.040      Min.   :0.4600      Min.   : 9.000
1st Qu.:3.250      1st Qu.:0.5500      1st Qu.: 9.325
Median :3.335      Median :0.5800      Median : 9.500
Mean   :3.337      Mean   :0.6836      Mean   : 9.742
3rd Qu.:3.415      3rd Qu.:0.7875      3rd Qu.: 9.875

```

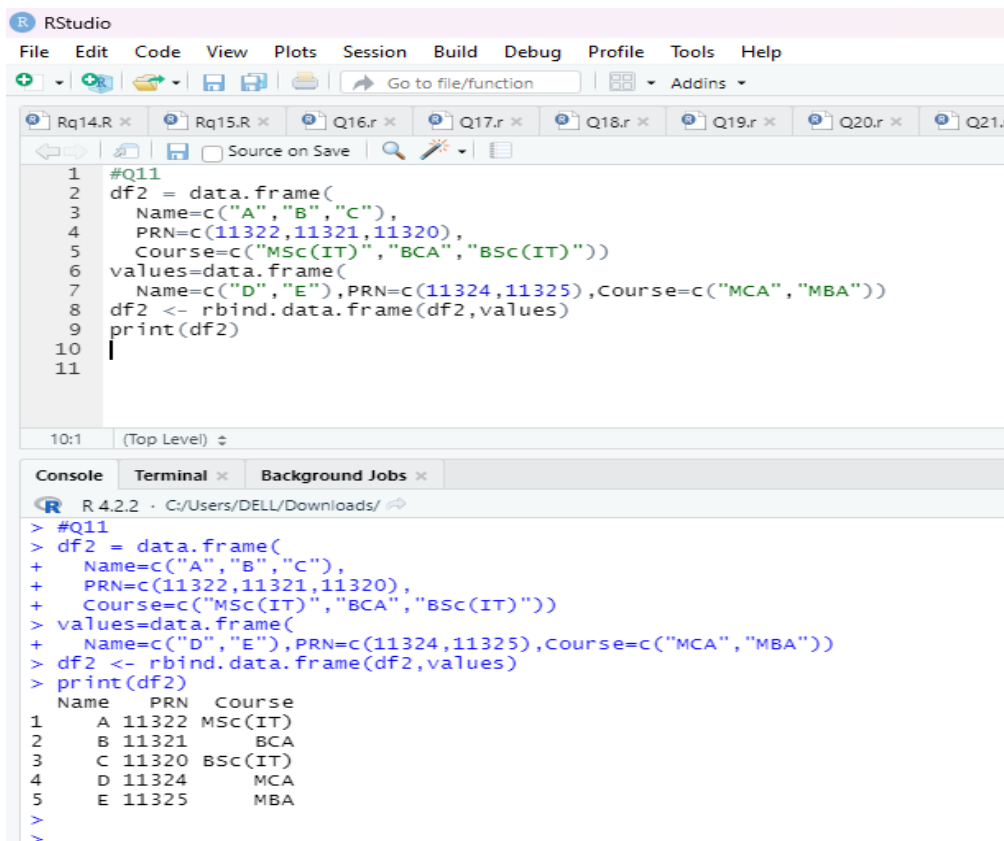
10. To extract first two rows from a given data frame

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Rq14.R Rq15.R Rq16.R Rq17.R Rq18.R Rq19.R Rq20.R
Source on Save
1 #Q10
2 df1 = head(read.csv("winequality-red.csv"),10)
3 print(df1[1:3])
4
4:1 (Top Level)
Console Terminal Background Jobs
R 4.2.2 C:/Users/DELL/Downloads/
> #Q10
> df1 = head(read.csv("winequality-red.csv"),10)
> print(df1[1:3])
      ID      fixed.acidity    volatile.acidity
1      1              7.4              0.70
2      2              7.8              0.88
3      3              7.8              0.76
4      4             11.2              0.28
5      5              7.4              0.70
6      6              7.4              0.66
7      7              7.9              0.60
8      8              7.3              0.65
9      9              7.8              0.58
10     10              7.5              0.50
>
> |

```

11. To add new row(s) to an existing data frame.



The screenshot shows the RStudio interface. The script editor contains the following code:

```

1 #Q11
2 df2 = data.frame(
3   Name=c("A","B","C"),
4   PRN=c(11322,11321,11320),
5   Course=c("MSc(IT)","BCA","BSc(IT)")
6 values=data.frame(
7   Name=c("D","E"),PRN=c(11324,11325),Course=c("MCA","MBA"))
8 df2 <- rbind.data.frame(df2,values)
9 print(df2)
10
11

```

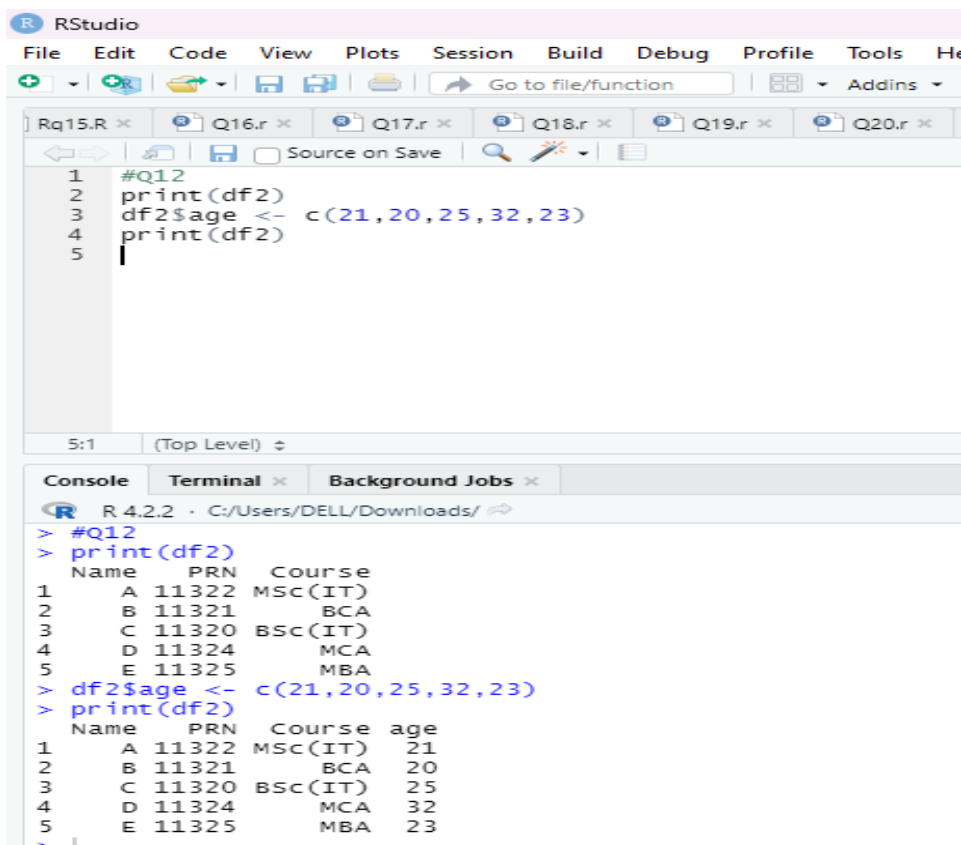
The console output shows the result of the code execution:

```

> #Q11
> df2 = data.frame(
+   Name=c("A","B","C"),
+   PRN=c(11322,11321,11320),
+   Course=c("MSc(IT)","BCA","BSc(IT)")
+ values=data.frame(
+   Name=c("D","E"),PRN=c(11324,11325),Course=c("MCA","MBA"))
> df2 <- rbind.data.frame(df2,values)
> print(df2)
  Name    PRN Course
1    A 11322 MSc(IT)
2    B 11321   BCA
3    C 11320 BSc(IT)
4    D 11324   MCA
5    E 11325   MBA

```

12. To add new Columns(s) to an existing data frame



The screenshot shows the RStudio interface. The script editor contains the following code:

```

1 #Q12
2 print(df2)
3 df2$age <- c(21,20,25,32,23)
4 print(df2)
5

```

The console output shows the result of the code execution:

```

> #Q12
> print(df2)
  Name    PRN Course
1    A 11322 MSc(IT)
2    B 11321   BCA
3    C 11320 BSc(IT)
4    D 11324   MCA
5    E 11325   MBA
> df2$age <- c(21,20,25,32,23)
> print(df2)
  Name    PRN Course age
1    A 11322 MSc(IT)  21
2    B 11321   BCA   20
3    C 11320 BSc(IT)  25
4    D 11324   MCA   32
5    E 11325   MBA   23

```


15. To replace NA values with 3 in a given data frame.

The screenshot shows the RStudio interface. The source editor contains the following code:

```
#Q15
print(df2)
df2[is.na(df2)] = 3
print(df2)
```

The console output shows the data frame before and after the replacement:

```
> #Q15
> print(df2)
  Name Course age
1    A  MSc(IT) 21
3    C  BSc(IT) 25
5    E    MBA  23

> df2[is.na(df2)] = 3
> print(df2)
  Name Course age
1    A  MSc(IT) 21
3    C  BSc(IT) 25
5    E    MBA  23
```

16. Create your own .csv file and read the data.

The screenshot shows the RStudio interface with the following code in the source editor:

```
#Q16
df2 = data.frame(
  Name=c("A","B","C","D","E"),
  PRN=c(11322,11321,11320,11324,11325),
  Course=c("MSc(IT)","BCA","BSc(IT)","MCA","MBA"))
print(df2)
write.csv(df2,"output.csv",row.names = FALSE)
```

The console output shows the data frame:

```
> #Q16
> df2 = data.frame(
+   Name=c("A","B","C","D","E"),
+   PRN=c(11322,11321,11320,11324,11325),
+   Course=c("MSc(IT)","BCA","BSc(IT)","MCA","MBA"))
> print(df2)
  Name PRN Course
1    A 11322 MSc(IT)
2    B 11321  BCA
3    C 11320 BSc(IT)
4    D 11324  MCA
5    E 11325  MBA

> write.csv(df2,"output.csv",row.names = FALSE)
>
```

Overlaid on the RStudio console is a screenshot of an Excel spreadsheet titled "Output.csv - Excel". The spreadsheet contains the following data:

Name	PRN	Course
A	11322	MSc(IT)
B	11321	BCA
C	11320	BSc(IT)
D	11324	MCA
E	11325	MBA

17. Perform functions like `print(is.data.frame(data))`, `print(ncol(data))` and `print(nrow(data))` and observe the output

```

1 #Q17
2 setwd("C:/Users/DELL/Downloads")
3 df1 = head(read.csv("winequality-red.csv"),50)
4 print(is.data.frame(df1))
5 print(ncol(df1))
6 print(nrow(df1))
7

```

```

> #Q17
> setwd("C:/Users/DELL/Downloads")
> df1 = head(read.csv("winequality-red.csv"),50)
> print(is.data.frame(df1))
[1] TRUE
> print(ncol(df1))
[1] 13
> print(nrow(df1))
[1] 50
>

```

18. Perform functions like `min`, `max`, `avg`, `mean`, `median` and `mode` and particular datacolumns

```

1 #Q18
2 setwd("C:/Users/DELL/Downloads")
3 df1 = read.csv("winequality-red.csv")
4 print(summary(df1))
5
6

```

```

R 4.2.2 - C:/Users/DELL/Downloads/
> #Q18
> setwd("C:/Users/DELL/Downloads")
> df1 = read.csv("winequality-red.csv")
> print(summary(df1))

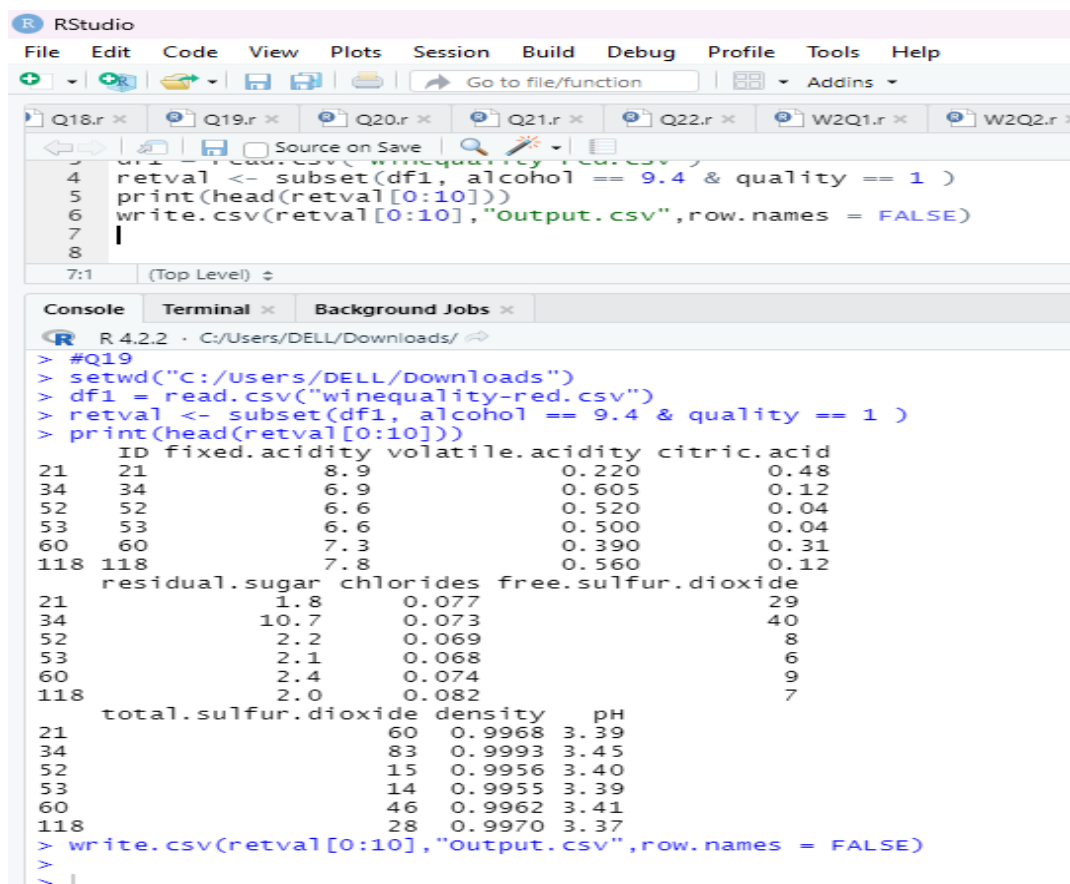
```

min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	7.00	14.00	15.89	21.00	72.00	3
0.00	22.00	38.00	46.45	62.00	289.00	3
0.9901	0.9956	0.9968	0.9967	0.9978	1.0037	2

min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.740	3.210	3.310	3.311	3.400	4.010
0.3300	0.5500	0.6200	0.6581	0.7300	2.0000
8.40	9.50	10.20	10.42	11.10	14.90

min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.0000	1.0000	0.5347	1.0000	1.0000

19. Create a logical subset of the data and write it in a new file.



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for file operations and a search bar. The source editor shows the following R code:

```

4   retval <- subset(df1, alcohol == 9.4 & quality == 1 )
5   print(head(retval[0:10]))
6   write.csv(retval[0:10], "Output.csv", row.names = FALSE)
7
8

```

The console output shows the execution of the code, including the reading of the CSV file and the printing of the first 10 rows of the subsetted data:

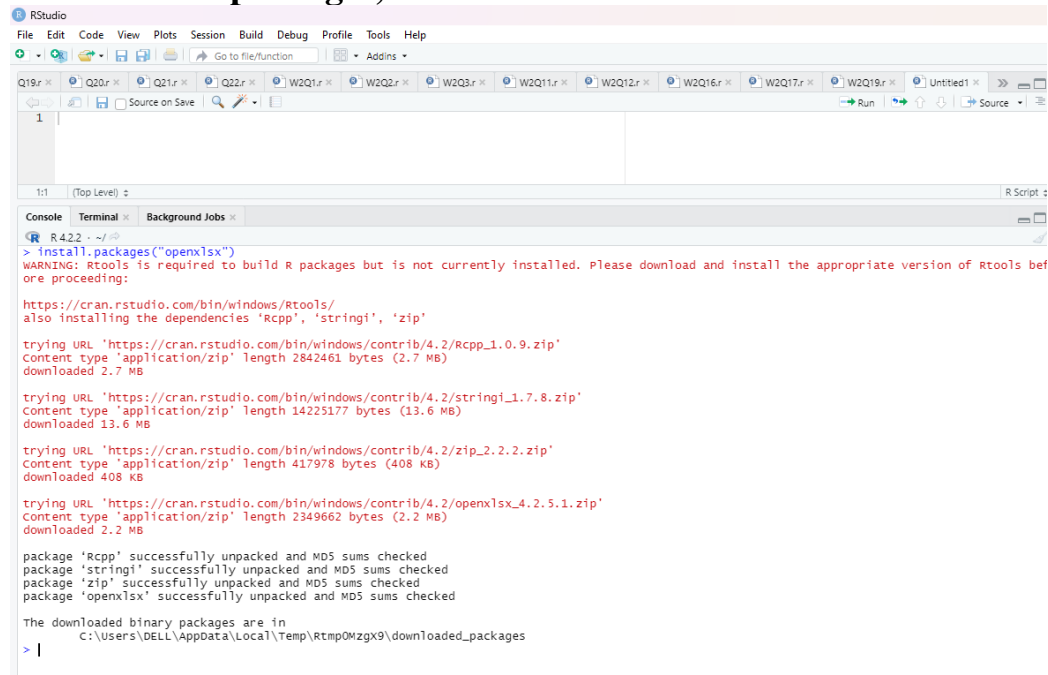
```

> #Q19
> setwd("C:/Users/DELL/Downloads")
> df1 = read.csv("winequality-red.csv")
> retval <- subset(df1, alcohol == 9.4 & quality == 1 )
> print(head(retval[0:10]))
  ID fixed.acidity volatile.acidity citric.acid
21  21           8.9             0.220      0.48
34  34           6.9             0.605      0.12
52  52           6.6             0.520      0.04
53  53           6.6             0.500      0.04
60  60           7.3             0.390      0.31
118 118          7.8             0.560      0.12
  residual.sugar chlorides free.sulfur.dioxide
21           1.8      0.077                  29
34          10.7      0.073                  40
52           2.2      0.069                   8
53           2.1      0.068                   6
60           2.4      0.074                   9
118          2.0      0.082                   7
  total.sulfur.dioxide density    pH
21           60 0.9968 3.39
34           83 0.9993 3.45
52           15 0.9956 3.40
53           14 0.9955 3.39
60           46 0.9962 3.41
118          28 0.9970 3.37
> write.csv(retval[0:10], "Output.csv", row.names = FALSE)
>

```

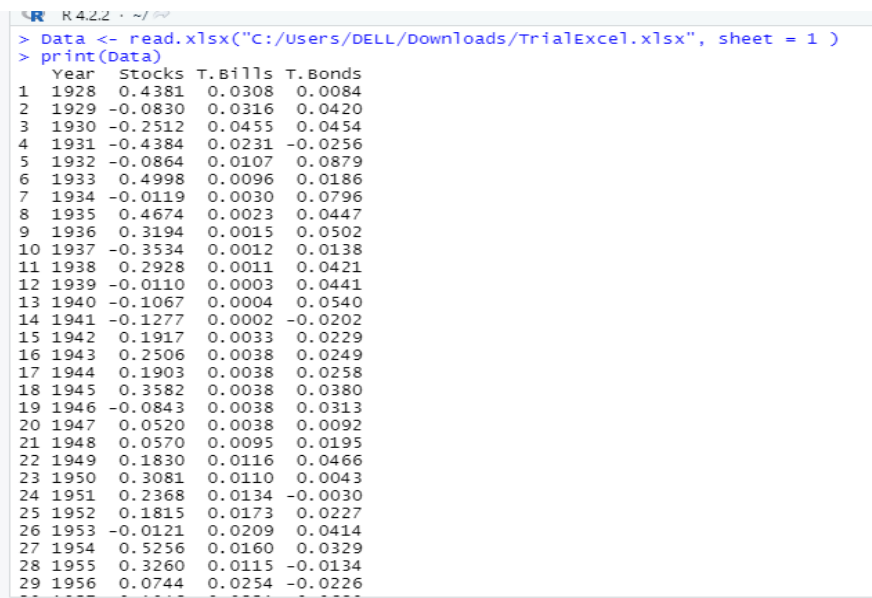

Week 3

1. To install xlsx packages, with and without JRE



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Q19r x Q20r x Q21r x Q22r x W2Q1r x W2Q2r x W2Q3r x W2Q11r x W2Q12r x W2Q16r x W2Q17r x W2Q19r x Untitled1 x
1
1:1 (Top Level) R Script
Console Terminal Background Jobs
R 4.2.2 ~ /
> install.packages("openxlsx")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
also installing the dependencies 'Rcpp', 'stringi', 'zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/Rcpp_1.0.9.zip'
Content type 'application/zip' length 2842461 bytes (2.7 MB)
downloaded 2.7 MB
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/stringi_1.7.8.zip'
Content type 'application/zip' length 14225177 bytes (13.6 MB)
downloaded 13.6 MB
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/zip_2.2.2.zip'
Content type 'application/zip' length 417978 bytes (408 KB)
downloaded 408 KB
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/openxlsx_4.2.5.1.zip'
Content type 'application/zip' length 2349662 bytes (2.2 MB)
downloaded 2.2 MB
package 'Rcpp' successfully unpacked and MD5 sums checked
package 'stringi' successfully unpacked and MD5 sums checked
package 'zip' successfully unpacked and MD5 sums checked
package 'openxlsx' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
c:\Users\DELL\AppData\Local\Temp\Rtmp0Wzgx9\downloaded_packages
> |
```

2. To open xlsx workbook or sheet.



```
R 4.2.2 ~ /
> Data <- read.xlsx("C:/Users/DELL/Downloads/TrialExcel.xlsx", sheet = 1 )
> print(Data)
  Year Stocks T.Bills T.Bonds
1 1928  0.4381  0.0308  0.0084
2 1929 -0.0830  0.0316  0.0420
3 1930 -0.2512  0.0455  0.0454
4 1931 -0.4384  0.0231 -0.0256
5 1932 -0.0864  0.0107  0.0879
6 1933  0.4998  0.0096  0.0186
7 1934 -0.0119  0.0030  0.0796
8 1935  0.4674  0.0023  0.0447
9 1936  0.3194  0.0015  0.0502
10 1937 -0.3534  0.0012  0.0138
11 1938  0.2928  0.0011  0.0421
12 1939 -0.0110  0.0003  0.0441
13 1940 -0.1067  0.0004  0.0540
14 1941 -0.1277  0.0002 -0.0202
15 1942  0.1917  0.0033  0.0229
16 1943  0.2506  0.0038  0.0249
17 1944  0.1903  0.0038  0.0258
18 1945  0.3582  0.0038  0.0380
19 1946 -0.0843  0.0038  0.0313
20 1947  0.0520  0.0038  0.0092
21 1948  0.0570  0.0095  0.0195
22 1949  0.1830  0.0116  0.0466
23 1950  0.3081  0.0110  0.0043
24 1951  0.2368  0.0134 -0.0030
25 1952  0.1815  0.0173  0.0227
26 1953 -0.0121  0.0209  0.0414
27 1954  0.5256  0.0160  0.0329
28 1955  0.3260  0.0115 -0.0134
29 1956  0.0744  0.0254 -0.0226
```

3. To use different argument settings while reading from excel sheet, pleaserefer the below command

```

> Data1 <- read.xlsx("C:/Users/DELL/Downloads/TrialExcel.xlsx", sheet = 1, startRow = 2)
> print(Data1)
      1928 0.43809999999999999 3.080000000000000001E-2
1 1929      -0.0830      0.0316
2 1930      -0.2512      0.0455
3 1931      -0.4384      0.0231
4 1932      -0.0864      0.0107
5 1933      0.4998      0.0096
6 1934      -0.0119      0.0030
7 1935      0.4674      0.0023
8 1936      0.3194      0.0015
9 1937      -0.3534      0.0012
10 1938      0.2928      0.0011
11 1939      -0.0110      0.0003
12 1940      -0.1067      0.0004
13 1941      -0.1277      0.0002
14 1942      0.1917      0.0033
15 1943      0.2506      0.0038
16 1944      0.1903      0.0038
17 1945      0.3582      0.0038
18 1946      -0.0843      0.0038
19 1947      0.0520      0.0038
20 1948      0.0570      0.0095
21 1949      0.1830      0.0116
22 1950      0.3081      0.0110
23 1951      0.2368      0.0134
24 1952      0.1815      0.0173
25 1953      -0.0121      0.0209
26 1954      0.5256      0.0160
27 1955      0.3260      0.0115
28 1956      0.0744      0.0254

```

```
R 4.2.2 ~ /> Data2 <- read.xlsx("c:/Users/DELL/Downloads/TrialExcel.xlsx", sheet = 1, startRow = 2, colNames = FALSE, rowNames = FALSE)
> print(Data2)
```

	x1	x2	x3	x4
1	1928	0.4381	0.0308	0.0084
2	1929	-0.0830	0.0316	0.0420
3	1930	-0.2512	0.0455	0.0454
4	1931	-0.4384	0.0231	-0.0256
5	1932	-0.0864	0.0107	0.0879
6	1933	0.4998	0.0096	0.0186
7	1934	-0.0119	0.0030	0.0796
8	1935	0.4674	0.0023	0.0447
9	1936	0.3194	0.0015	0.0502
10	1937	-0.3534	0.0012	0.0138
11	1938	0.2928	0.0011	0.0421
12	1939	-0.0110	0.0003	0.0441
13	1940	-0.1067	0.0004	0.0540
14	1941	-0.1277	0.0002	-0.0202
15	1942	0.1917	0.0033	0.0229
16	1943	0.2506	0.0038	0.0249
17	1944	0.1903	0.0038	0.0258
18	1945	0.3582	0.0038	0.0380
19	1946	-0.0843	0.0038	0.0313
20	1947	0.0520	0.0038	0.0092
21	1948	0.0570	0.0095	0.0195
22	1949	0.1830	0.0116	0.0466
23	1950	0.3081	0.0110	0.0043
24	1951	0.2368	0.0134	-0.0030
25	1952	0.1815	0.0173	0.0227
26	1953	-0.0121	0.0209	0.0414
27	1954	0.5256	0.0160	0.0329
28	1955	0.3260	0.0115	-0.0134

```
> Data3 <- read.xlsx("C:/Users/DELL/Downloads/TrialExcel.xlsx", sheet = 1, startRow = 2, colNames = FALSE, rowNames = FALSE, detectDates = TRUE, skipEmptyRows = TRUE, rows = c(1,2,5,8), cols = c(1,2,4), check.names = TRUE, sep.names = "_", namedRegion = NULL, na.strings = "NA", fillMergedCells = TRUE)
> print(Data3)
```

	X1	X2	X3
1	1928	0.4381	0.0084
2	1931	-0.4384	-0.0256
3	1934	-0.0119	0.0796

4. Create your own function and implement it on a particular column/rowon excel sheet data.

```

1 total_bill_bonds1 <- function(df){
2   total = df$T.Bills + df$T.Bonds
3   return(total)
4 }
5 Data4 <- head(read.xlsx("C:/Users/DELL/Downloads/TrialExcel.xlsx",
6   Data$total_bill_bonds1 <- total_bill_bonds1(Data4)
7   print(Data4)
8   write.xlsx(Data4, "output1212.xlsx")
9
10

```

The console shows the output of the function applied to the data, displaying a table with columns for Year, Stocks, T.Bills, T.Bonds, and total_bill_bonds1.

Year	Stocks	T.Bills	T.Bonds	total_bill_bonds1
1928	43.81%	3.08%	0.84%	3.92%
1929	-8.30%	3.16%	4.20%	7.36%
1930	-25.12%	4.55%	4.54%	9.09%
1931	-43.84%	2.31%	-2.56%	-0.25%
1932	-8.64%	1.07%	8.79%	9.86%
1933	49.98%	0.96%	1.86%	2.82%
1934	-1.19%	0.30%	7.96%	8.26%
1935	46.74%	0.23%	4.47%	4.70%
1936	31.94%	0.15%	5.02%	5.17%
1937	-35.34%	0.12%	1.38%	1.50%
1938	29.28%	0.11%	4.21%	4.32%
1939	-1.10%	0.03%	4.41%	4.44%
1940	-10.67%	0.04%	5.40%	5.44%
1941	-12.77%	0.02%	-2.02%	-2.00%
1942	19.17%	0.33%	2.29%	2.62%
1943	25.06%	0.38%	2.49%	2.87%
1944	19.03%	0.38%	2.58%	2.96%
1945	35.82%	0.38%	3.80%	4.18%
1946	-8.43%	0.38%	3.13%	3.51%
1947	5.20%	0.38%	0.92%	1.30%
1948	5.70%	0.95%	1.95%	2.90%
1949	18.30%	1.16%	4.66%	5.82%
1950	30.81%	1.10%	0.43%	1.53%
1951	23.68%	1.34%	-0.30%	1.04%
1952	18.15%	1.73%	2.27%	4.00%
1953	-1.21%	2.09%	4.14%	6.23%
1954	52.56%	1.60%	3.29%	4.89%
1955	32.60%	1.15%	-1.34%	-0.19%