PYTHON LAB RECORD

WEEK1: Input and output
a)
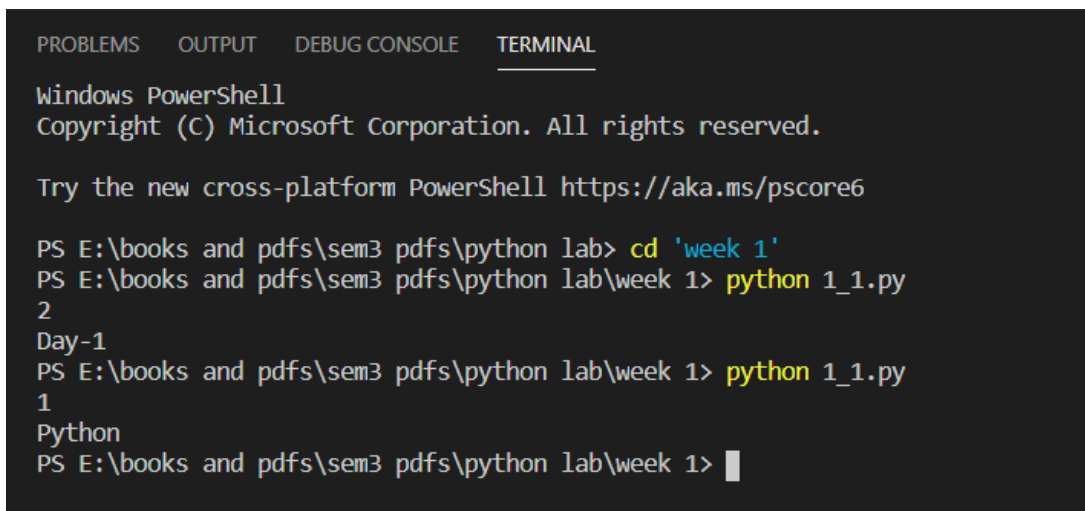**AIM:**
Print the "Python" for 1, print "Day - 1" for 2. By changing the variable "look" for each statement.

CODE

```python
i=int(input())
''' Taking user input and checking its value to make look=python if input=1
else if input=2 make look=day-1 and print look'''
if i==1: look="Python"
elif i==2: look="Day-1"
print(look)
```

OUTPUT:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\books and pdfs\sem3 pdfs\python lab> cd 'week 1'
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_1.py
2
Day-1
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_1.py
1
Python
PS E:\books and pdfs\sem3 pdfs\python lab\week 1>
```

b)
AIM: Create a variable "number" and assign an Integer to the number. Check the assigned Integer is "Positive" or "Negative".

CODE:

```python
n=int((input()))
#take user input and check if it's a positive or negative number by comparing
values
print(n,"is positive") if n>=0 else print(n,"is negative")
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_2.py
45
45 is positive
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_2.py
-2
-2 is negative
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_2.py
0
0 is positive
PS E:\books and pdfs\sem3 pdfs\python lab\week 1>
```

c)
Write a program to find the largest element among three Numbers.

```python
a,b,c=map(int,input().split())

#take 3 user inputs and print greatest among the three

print(a ,"is the greatest number among ",a,b,c) if a>b and a>c else print(b,"is
the greatest number among ",a,b,c) if b>c and b>a else print(c,"is the greatest
number among ",a,b,c)
```

OUTPUT:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_3.py
56 98 45
98 is the greatest number among  56 98 45
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_3.py
-2 -5 -1
-1 is the greatest number among  -2 -5 -1
PS E:\books and pdfs\sem3 pdfs\python lab\week 1>
```

d)
AIM: Write a program to print the sum of all the even number in the range 1 - 50 and print the even sum.

```
c=0

#printing the sum of all even numbers in range 1 to 50 using a for loop

for i in range(1,51):
    if i%2==0:
        c+=i
print(c)
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_4.py
650
```
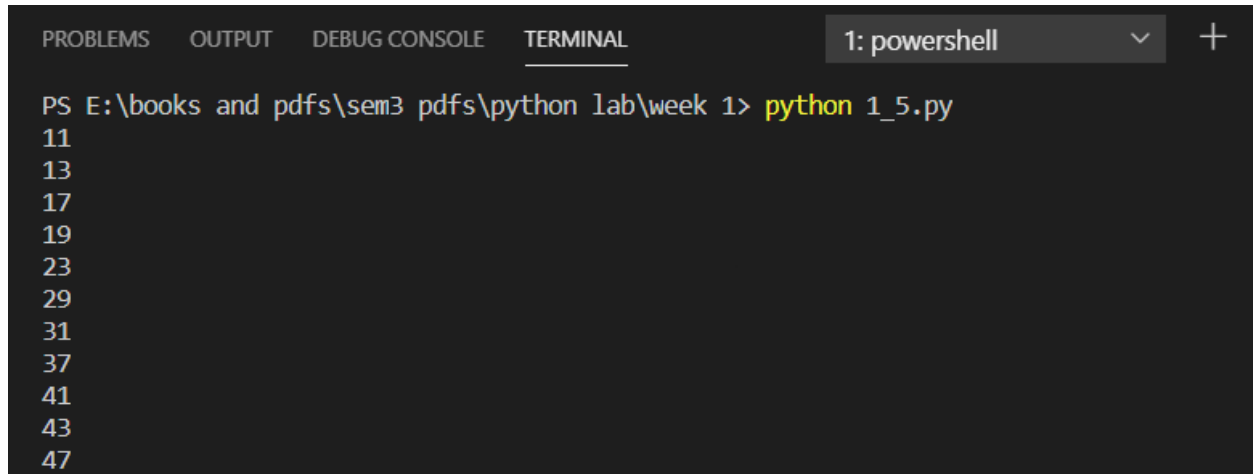
e)
AIM: Write a program to display all prime numbers within an interval of 20 and 50.

```
def prime(n):
        #checking if a number is prime or not. if any number between 2,n//2 is
    divisible by n then it is not a prime number.

    for i in range(2,n//2):
        if n%i==0:
            return 0
    return 1

# printing all prime numbers in range 10 to 51
for i in range(10,51):
    if prime(i)==1:
print(i)
```
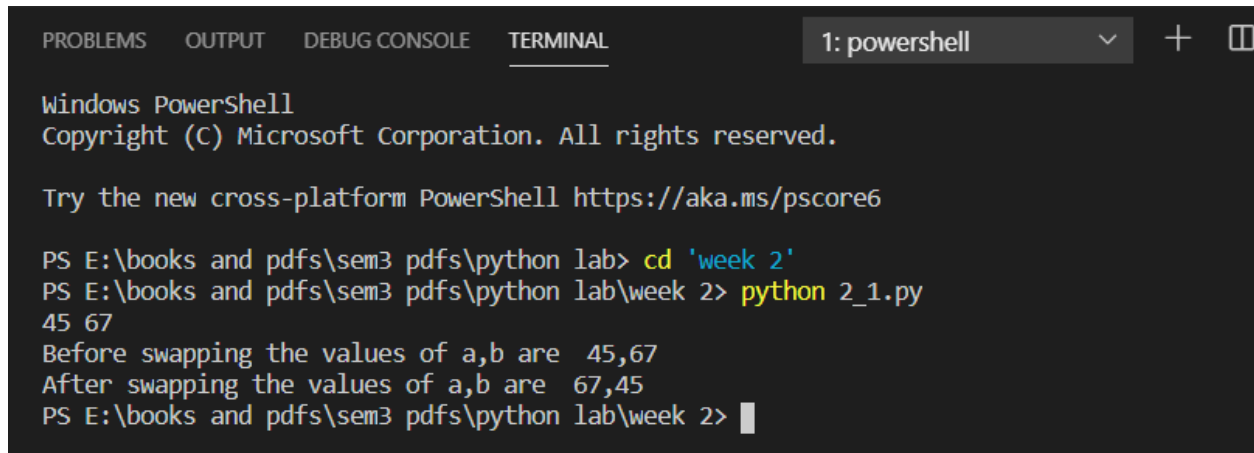
OUTPUT:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL          1: powershell          ∨    +

PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_5.py
11
13
17
19
23
29
31
37
41
43
47
```

WEEK 2 : Variables and Functions

a)AIM:Write a program to swap two numbers without using a temporary variable.

Code:

```python
a,b=input().split()
# values of a and b before swapping are printed
print("Before swapping the values of a,b are ",a+','+b)
a,b=b,a
# a and b are swapped and printed
print("After swapping the values of a,b are ",a+','+b)
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL              1: powershell          ∨    +  ▯

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\books and pdfs\sem3 pdfs\python lab> cd 'week 2'
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> python 2_1.py
45 67
Before swapping the values of a,b are  45,67
After swapping the values of a,b are  67,45
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> ▮
```
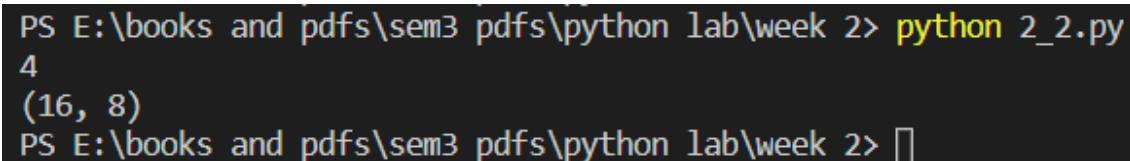
b)AIM: Write a program to define a function with multiple return values.

Code:

```python
def func(n):
    # return 2 values to the calling function
    return n*n,n+n

# Take user input and send it to a function and receive 2 values and print them
n=int(input())
print(func(n))
```

OUTPUT:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> python 2_2.py
4
(16, 8)
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> ▯
```

c) AIM: Write a program which creates an adder given a value (Use only lambda)

CODE:

```python
adder=lambda n,m:n+m
# taking user input using map function and finding sum of values using lambda
function
n,m=map(int,input().split())
print(adder(n,m) ,"is the adder of",n,"and",m,"using lambda function")
```

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> python 2_3.py
34 56
90 is the adder of 34 and 56 using lambda funtions
PS E:\books and pdfs\sem3 pdfs\python lab\week 2>
```

d)
Aim:Write a program to define a function using default arguments.

```python
def my_func(x,n="User"): #defining a function with default arguments
    print(x,n)

a,b=map(str,input().split())
my_func(a,b)   #calling function with 2 arguments
my_func(a)    # calling function with only 1 argument
```

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> python 2_4.py
gvp college
gvp college
gvp User
PS E:\books and pdfs\sem3 pdfs\python lab\week 2>
```

WEEK 3: Loops and conditionals

a)Write a program to print the following patterns using loop:

*
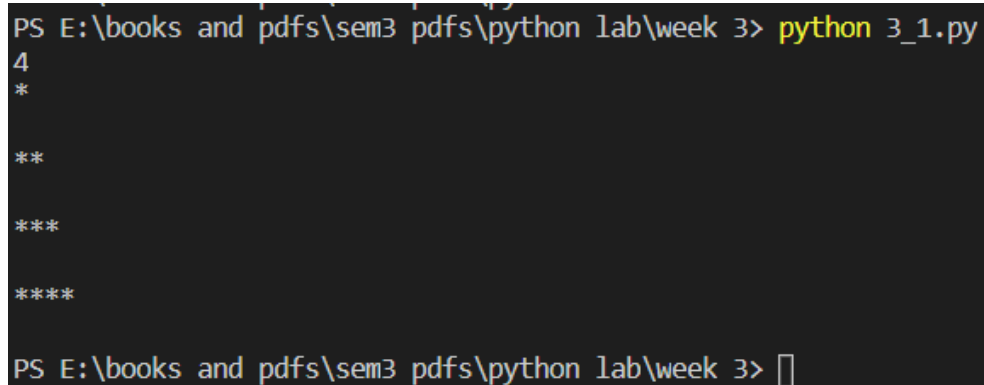
**

***

****

Code:

```
n=int(input())
# printing the given pattern using 1 for loop and * operator
for i in range(1,n+1):
    print(i* '*','\n')
```
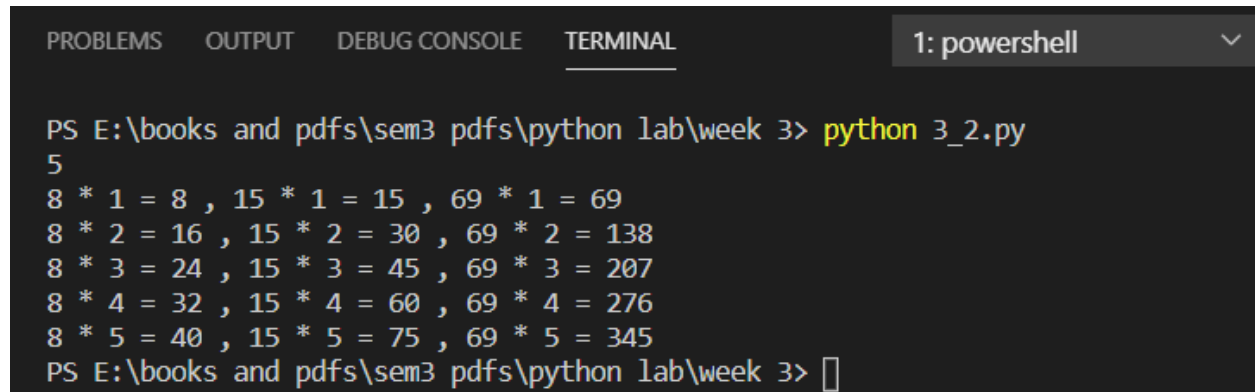
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 3> python 3_1.py
4
*

**

***

****

PS E:\books and pdfs\sem3 pdfs\python lab\week 3>
```

b)
Aim:Write a program to print multiplication tables of 8, 15, 69.

Code:

```
n=int(input())
# printing multiplication tables of 8,15,69 at the same time
for i in range(1,n+1):
    print(8,'*',i,'=',8*i,',',15,'*',i,'=',15*i,',',69,'*',i,'=',69*i)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL              1: powershell          ∨

PS E:\books and pdfs\sem3 pdfs\python lab\week 3> python 3_2.py
5
8 * 1 = 8 , 15 * 1 = 15 , 69 * 1 = 69
8 * 2 = 16 , 15 * 2 = 30 , 69 * 2 = 138
8 * 3 = 24 , 15 * 3 = 45 , 69 * 3 = 207
8 * 4 = 32 , 15 * 4 = 60 , 69 * 4 = 276
8 * 5 = 40 , 15 * 5 = 75 , 69 * 5 = 345
PS E:\books and pdfs\sem3 pdfs\python lab\week 3> ▯
```
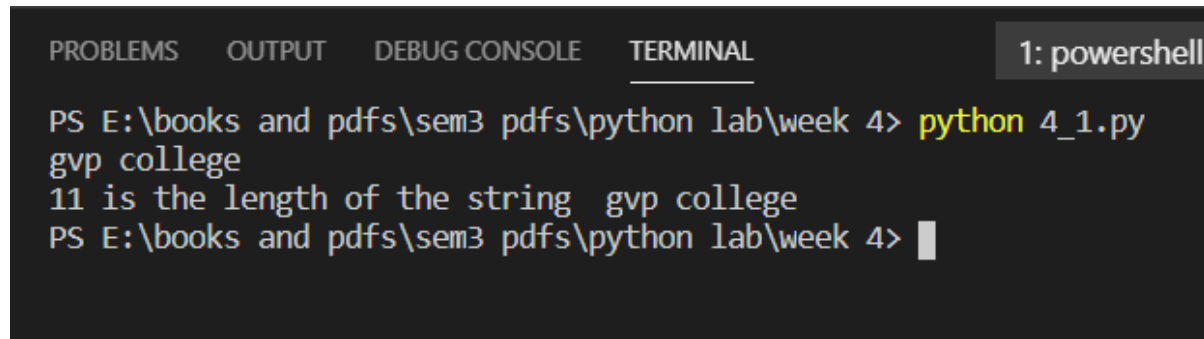
WEEK 4: Strings

a)Aim: Write a program to find the length of the string without using any library functions.

```
str1=input()
c=0
# for loop to find length of string
for i in str1:
    c+=1
    # if character is present then c is incremented by 1
print(c,"is the length of the string ",str1)
```

Output:

b)Aim: Write a program to check if two strings are anagrams or not.
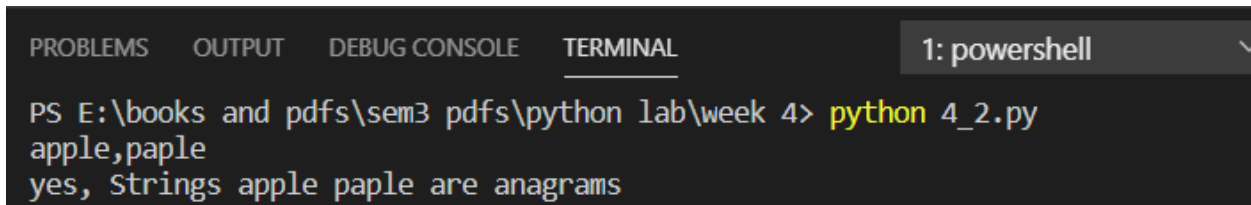
Code 1:

```python
def fun(a,b):
# if every character of a is present in b then return 1 . else return 0
    for i in a:
        if i not in b:
            return 0
    return 1
# taking 2 user inputs
a,b=input().split(',')
# if function value is 1 then they are anagrams else they are not anagrams
print("yes, Strings "+a,b+ " are anagrams ") if fun(a,b) else print("no,
Strings "+a,b +" are not anagrams ")
```

Code 2:

```python
if sorted(a)==sorted(b):print("yes, Strings "+a,b+ " are anagrams ")
else:print("no, Strings "+a,b +" are not anagrams ")
```

Output:


```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                1: powershell

PS E:\books and pdfs\sem3 pdfs\python lab\week 4> python 4_2.py
apple,paple
yes, Strings apple paple are anagrams
```

c)
**Aim: Write a program to check if a substring is present in a
given string or not.**

Code 1:

```python
def fun(a,b):
    #if b is present in a then return 1 else return 0

    if b in a:
        return 1
    return 0
a,b=input().split(',')
print("yes,Substring "+b+" is present in substring "+a) if fun(a,b) else
print("no,Substring "+b+" is not present in substring "+a)
```

Code 2:

```python
import re
a,b=input().split(',')
# if b is present in a then re.search(b,a) will be 1 else 0
if re.search(b,a):
    print("yes,Substring "+b+" is present in substring "+a)
else:
    print("no,Substring "+b+" is not present in substring "+a)
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL          1: powershell        ∨   +

PS E:\books and pdfs\sem3 pdfs\python lab\week 4> python 4_3.py
gvp college,gvp
yes,Substring gvp  is present in substring gvp college
PS E:\books and pdfs\sem3 pdfs\python lab\week 4>
```

WEEK 5 : Lists
a)Aim: Write a program to perform the given operations on a list:
i. add ii. insert iii. slicing

Code:

```python
li1=['Gayathri',2,'Rajahmundry','Vizag',5,7]
li2=['Kavitha',13,'Hyderabad','vizag',6,12]
# taking 2 lists li1 and li2
print(li1+li2)
li1.append(234) # appending both the lists
li1.insert(1,13) # inserting a value in list 1 at a particular position
print(li1)
print(li1[1:4]) # applying slicing operation on list1
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> python 5_1.py
['Gayathri', 2, 'Rajahmundry', 'Vizag', 5, 7, 'Kavitha', 13, 'Hyderabad', 'vizag', 6, 12]
['Gayathri', 13, 2, 'Rajahmundry', 'Vizag', 5, 7, 234]
[13, 2, 'Rajahmundry']
```

b)Aim: Write a program to perform any 5 built-in functions by taking any list.

Code:

```python
li1=['Gayathri',2,'Rajahmundry','Vizag',5,7]
li2=['Kavitha',13,'Hyderabad','vizag',6,12]

# taking 2 lists li1 and li2
print(li2.pop(2)) # popping a value 2 from list 2
```

```
li1.clear() # clearing list 1 and printing it
print(li1)
print(li2)
li2.remove(12) # removing a value from li2
print(li2)
li2.reverse() # reversing list2
print(li2)
li1=li2.copy() # copying list 2 to list 1
print(li1)
li1.append('asdbj') # appending a value into list1
print(li1)
del li1 # deleting list 1
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> python 5_2.py
Hyderabad
[]
['Kavitha', 13, 'vizag', 6, 12]
['Kavitha', 13, 'vizag', 6]
[6, 'vizag', 13, 'Kavitha']
[6, 'vizag', 13, 'Kavitha']
[6, 'vizag', 13, 'Kavitha', 'asdbj']
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> []
```

c)Aim: Write a program to get a list of even numbers from a given list of numbers.(use only comprehensions)
Code:

```
li=list(map(int,input().split()))
# taking a list of numbers from user i=using map and list functions
print([i for i in li if i%2==0])
# using list comprehension to find list of even numbers.
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> python 5_3.py
3 45 2 7 8 80 75 32 11 24
[2, 8, 80, 32, 24]
PS E:\books and pdfs\sem3 pdfs\python lab\week 5>
```

WEEK 6: Tuples

a) Aim: Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenate tuples.

```python
t1=tuple((input("name "),int(input("Age ")),input("address "),input("college ")))

t2=tuple((input("name "),int(input("Age ")),input("address "),input("college ")))
# taking 2 tuples as input from user using tuple and input functions

print(t1+t2) # concatenating the tuples and printing the result
```

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 6> python 6_1.py
name Ramu
Age 30
address kommadi
college gvp
name ravi
Age 28
address vizag
college AU
('Ramu', 30, 'kommadi', 'gvp', 'ravi', 28, 'vizag', 'AU')
PS E:\books and pdfs\sem3 pdfs\python lab\week 6>
```

b) Aim: Write a program to return the top 'n' most frequently occurring chars and their respective counts.

e.g. aaaaaabbbbcccc, 2 should return [(a 6) (b 4)]
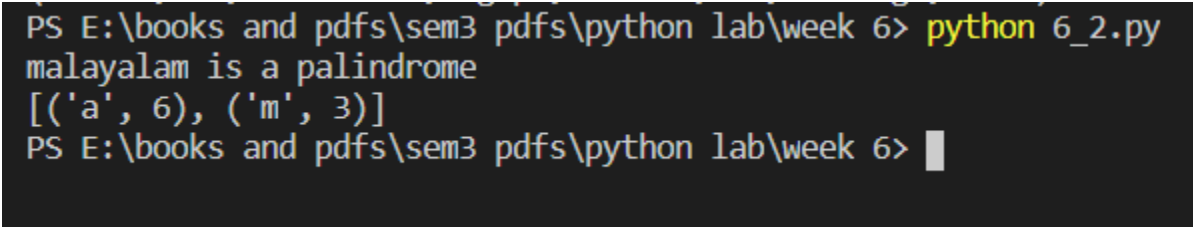
Code 1:

```python
a=input()
n=int(input())
# taking user input and finding the count of all characters and making a list
of it
li=[(i,a.count(i)) for i in a]
li=list(dict.fromkeys(li))
li= sorted(li, key=lambda tup: tup[1],reverse=True)
# using a lambda function sorting the list in reverse order
print(li[0:n]) # using slicing operation to print most n
```

Code 2:

```python
from collections import Counter  # importing counter from collections package
counts = Counter(input()) # counting all characters using count function
print(counts.most_common(2)) # printing most common n char and their occurance
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 6> python 6_2.py
malayalam is a palindrome
[('a', 6), ('m', 3)]
PS E:\books and pdfs\sem3 pdfs\python lab\week 6>
```

WEEK 7: Sets

a) **Aim:** Write a program to count the number of vowels in a string (No control flow allowed).

Code 1:

```python
a=input()
b=set("aeiouAEIOU")
print(sum([1 for i in a if i in b]))
# using list comprehension to find number of vowels present in string a
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL          1: powershell
PS E:\books and pdfs\sem3 pdfs\python lab> cd 'week 7'
PS E:\books and pdfs\sem3 pdfs\python lab\week 7> python 7_1.py
Gayatri Vidhya Parishad college of engineering(A)
18
PS E:\books and pdfs\sem3 pdfs\python lab\week 7>
```

b)Aim: Write a program that displays which letters are present in both strings.

Code 1:

```python
a,b=set(input().split(','))
# using list comprehension to find common letters in both strings
print([i for i in a if i in b])
```

Code 2:

```python
a,b=set(input().split(','))
c = set(a).intersection(set(b))
# using intersection function to find common characters of both strings
print(c)
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 7> python 7_2.py
gvp college,gvp ground
['g', 'v', 'p', ' ', 'o', 'g']
PS E:\books and pdfs\sem3 pdfs\python lab\week 7>
```

c)Aim: Write a program to sort given list of strings in the order of their vowel counts.

Code:

```python
def vowelcount(n):
    # using list comprehension to find number of even characters
```

```python
    a=sum([1 for i in n if i in 'aeiouAEIOU'])
    return a

n=input().split()
# using sorted function to sort all strings in order of vowel count
print(sorted(n,key=vowelcount))
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 7> python 7_3.py
hello world
['world', 'hello']
PS E:\books and pdfs\sem3 pdfs\python lab\week 7> python 7_3.py
gvp college of engineering
['gvp', 'of', 'college', 'engineering']
PS E:\books and pdfs\sem3 pdfs\python lab\week 7>
```

WEEK 8: Dictionaries

a) Aim: Write a program to generate a dictionary that contains
   numbers (between 1 and n) in the form of (x, x*x).

Code 1:

```python
n=int(input())
# generating a dictionary i,i*i
print({i:i*i for i in range(n)})
```

Code 2:

```python
res=dict()
```

```python
for i in range(n):
    # using a dictionary variable and append i*i values into the dictionary
variable and then print it
    res[i]=i*i
print(res)
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_1.py
5
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
PS E:\books and pdfs\sem3 pdfs\python lab\week 8>
```

b) **Aim: Write a program to check if a given key exists in a dictionary or not.**

Code:

```python
n=int(input("number of key:value pairs in dictionary"))
d=dict(input("key and value ").split() for i in range(n))
k=input("key ")
# taking a dictionary as input from user
l={d[i] for i in d if i==k}
# storing value for given key and print it if present
if l:print("key is found in dictionary and its corresponding value is ",l)
else:print("key not found in dictionary")
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_2.py
number of key:value pairs in dictionary4
key and value hello 1
key and value hi 2
key and value morning 3
key and value afternoon 4
key hi
key is found in dictionary and its corresponding value is  {'2'}
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_2.py
number of key:value pairs in dictionary4
key and value hello 1
key and value hi 2
key and value morning 3
key and value afternoon 4
key evening
key not found in dictionary
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> █
```

c) **Aim: Write a program to add a new key-value pair to an existing dictionary.**

Code:

```python
n=int(input("number of key:value pairs in dictionary"))
d=dict(input("key and value ").split() for i in range(n))
# input a dictionary from user
d.update({"hello":"999"}) #update it by adding a new key value pair
print(d)
```

Output:
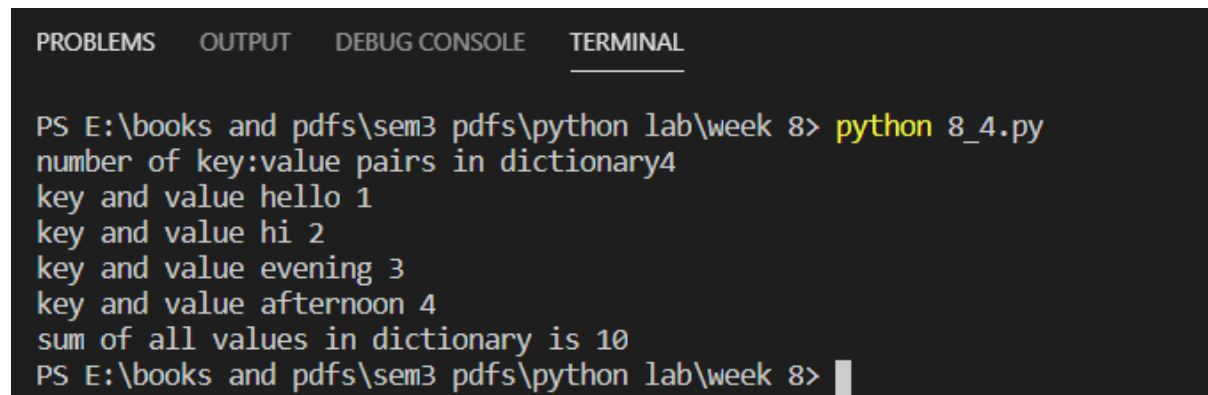
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_3.py
number of key:value pairs in dictionary4
key and value morning 1
key and value hi 2
key and value evening 3
key and value afternoon 4
{'morning': '1', 'hi': '2', 'evening': '3', 'afternoon': '4', 'hello': '999'}
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> █
```

d) **Aim: Write a program to sum all the items in a given dictionary.**

Code:

```
n=int(input("number of key:value pairs in dictionary"))
d=dict(input("key and value ").split() for i in range(n))
# input a dictionary from user
print("sum of all values in dictionary is",sum([int(d[i]) for i in d]))
# finding sum of all values in dictionary and print it
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_4.py
number of key:value pairs in dictionary4
key and value hello 1
key and value hi 2
key and value evening 3
key and value afternoon 4
sum of all values in dictionary is 10
PS E:\books and pdfs\sem3 pdfs\python lab\week 8>
```

WEEK 9: Files

a) Aim: a. Write a program to sort words in a file and put them in another file. The output file should have only lower case words, so any upper case words from source must be lowered. (Handle exceptions)

Code:

```
f1 = open("file1.txt",'r')
l = f1.read().split("\n")
lt = []
for i in l:
    for j in i.split():
        lt.append(j.lower())
        # appending all characters of file1 converting into lowercase
lt.sort() # sort all the words
```

```python
print(lt) # print all the words
f2 = open("output.txt", "w")
for i in lt:
    f2.write(str(i)+"\n") # write them to another file
f1.close()
f2.close() # close all file pointers
```

file1.txt:

```
≡ file1.txt   ×    ≡ file2.txt

week 9 >  ≡ file1.txt
    1    Python is a general-purpose coding language—which means that, unlike HTML, CSS, and JavaScript,
    2    it can be used for other types of programming and software development besides web development.
    3    That includes back end development, software development, data science and writing system scripts among other things.
    4    IT IS USED FOR:
    5    WEB DEVELOPMENT (SERVER-SIDE),
    6    SOFTWARE DEVELOPMENT,
    7    MATHEMATICS,
    8    SYSTEM SCRIPTING.
```

```
PS E:\books and pdfs\sem3 pdfs\python lab> cd 'week 9'
PS E:\books and pdfs\sem3 pdfs\python lab\week 9> python 9_1.py
['(server-side),', 'a', 'among', 'and', 'and', 'and', 'back', 'be', 'besides', 'can', 'coding', 'css,', 'data', 'development', 'development', 'development,', 'develo
pment,', 'development,', 'development.', 'end', 'for', 'for:', 'general-purpose', 'html,', 'includes', 'is', 'is', 'it', 'it', 'javascript,', 'languageâ€"which', 'ma
thematics,', 'means', 'of', 'other', 'other', 'programming', 'python', 'science', 'scripting.', 'scripts', 'software', 'software', 'software', 'system', 'system', 't
hat', 'that,', 'things.', 'types', 'unlike', 'used', 'used', 'web', 'web', 'writing']
PS E:\books and pdfs\sem3 pdfs\python lab\week 9> █
```

Output.txt:

```
week 9 >  ≡ output.txt
 1   (server-side),
 2   a
 3   among
 4   and
 5   and
 6   and
 7   back
 8   be
 9   besides
10   can
11   coding
12   css,
13   data
14   development
15   development
16   development,
17   development,
18   development,
19   development.
20   end
21   for
22   for:
23   general-purpose
24   html,
25   includes
26   is
27   is
28   it
29   it
30   javascript,
31   language—which
32   mathematics,
33   means
34   of
35   other
36   other
37   programming
38   python
39   science
40   scripting.
41   scripts
42   software
43   software
44   software
45   system
46   system
47   that
48   that,
49   things.
50   types
51   unlike
52   used
53   used
54   web
55   web
56   writing
57
```
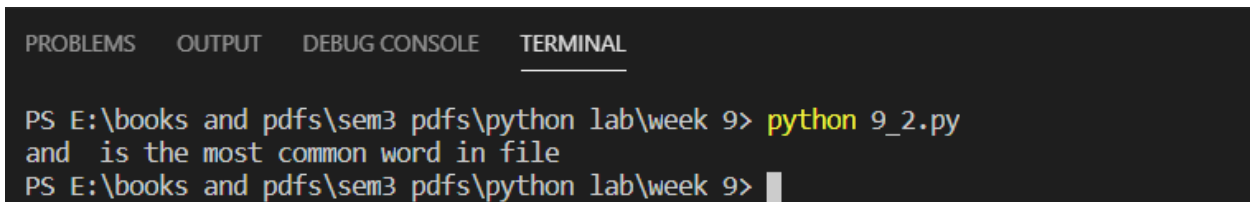
b) Aim: Write a program to find the most frequent words in a
   text.(read from a text file)

Code:

```python
from collections import Counter
li=[]
f=open("file1.txt","r")
for i in f:
    for j in i.split():
        # splitting all the words of text in file 1 and appending into a list
        if j!='\n' or j!='\t':
            li.append(j)
# using counter function to count the occurrence of each word
mo=Counter(li)
print(mo.most_common(1)[0][0]," is the most common word in file")
# printing most common word from file
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\books and pdfs\sem3 pdfs\python lab\week 9> python 9_2.py
and  is the most common word in file
PS E:\books and pdfs\sem3 pdfs\python lab\week 9>
```

WEEK 10: Classes

a) Aim:Write a Python class named Person with attributes name,
   age, weight (kgs), height (ft) and takes them through the
   constructor and exposes a method get_bmi_result() which
   returns one of "underweight", "healthy", "obese".

Code:

```python
class Person:
    def __init__(self,name,age,weight,height):
        self.name=name
        self.age=age
        self.weight=weight
        self.height=height
        # using init constructor to store values into variables
    def get_bmi_result(self):
        h=self.height/100    # dividing height of person by 100 to find bmi
        bmi=self.weight/(h**2)   # finding bmi of person
        if bmi<=18.5:  # if bmi<18.5 then return status under-weight
            return "\n---------\nYour
report\nname:{0}\nage:{1}\nweight:{2}\nheight:{3}\n STATUS:
Under-weight".format(self.name,self.age,self.weight,self.height)
        elif bmi>18.5 and bmi<25:
        # if bmi>18.5 and bmi<25 then return status healthy
            return "\n---------\nYour
report\nname:{0}\nage:{1}\nweight:{2}\nheight:{3}\n STATUS:
Healthy".format(self.name,self.age,self.weight,self.height)
        elif bmi>=25: # if bmi>= then return status overweight
            return "\n---------\nYour
report\nname:{0}\nage:{1}\nweight:{2}\nheight:{3}\n STATUS:
Over-weight".format(self.name,self.age,self.weight,self.height)


p1=Person(input("name: "),int(input("age ")),float(input("weight
")),float(input("height ")))
p1.get_bmi_result()  # input the details of person and find his bmi
print(Person.get_bmi_result(p1)) # print the bmi of p1
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\books and pdfs\sem3 pdfs\python lab\week 10> python 10_1.py
name: Ramu
age 21
weight 68
height 171


---------
Your report
name:Ramu
age:21
weight:68.0
height:171.0
 STATUS: Healthy
PS E:\books and pdfs\sem3 pdfs\python lab\week 10> █
```

b) **Aim:Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.**

Code:

```python
class circle:
    def __init__(self,radius):
        self.radius=radius
        # using init constructor to store radius
    def perimeter(self):
        return 2*3.14*self.radius
        # return perimeter of circle of given radius

    def area(self):
        return 3.14*self.radius*self.radius
        # return area of circle of given radius



c1=circle(float(input("radius of circle: ")))
c1.area()
# calling area and perimeter functions and printing the results
print("Area of circle is ",circle.area(c1))
c1.perimeter()
print("Perimeter of circle is ",circle.perimeter(c1))
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\books and pdfs\sem3 pdfs\python lab\week 10> python 10_2.py
radius of circle: 10.0
Area of circle is  314.0
Perimeter of circle is  62.800000000000004
PS E:\books and pdfs\sem3 pdfs\python lab\week 10> ▯
```

WEEK 11: Arrays
   a) Aim: Write a program to create, display, append, insert and
      reverse the order of the items in the array.

   Code:

```python
import numpy as np
arr=np.array(input().split())
# creating a numpy array with some values in it
print(np.append(arr,12))
# appending a value into array
arr=np.insert(arr,1,9)
# inserting a value at a particular index
print(arr)
print(np.flip(arr))
# reversing the numpy array
```

Output:
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 11> python 11_1.py
2 4 6 12 34 23 76 11 22
['2' '4' '6' '12' '34' '23' '76' '11' '22' '12']
['2' '9' '4' '6' '12' '34' '23' '76' '11' '22']
['22' '11' '76' '23' '34' '12' '6' '4' '9' '2']
PS E:\books and pdfs\sem3 pdfs\python lab\week 11> ▮
```

b) Aim:

Write a program to add, transpose and multiply two matrices.

Code:

```python
import numpy as np
m1 = np.array([[1,4,7],[2,5,8]])
m2 = np.array([[1,4,8],[2,3,6]])
print('array 1\n',m1)
print('array 2\n',m2)
print('\nMultiplication\n',np.multiply(m1,m2))
# using inbuilt numpy functions multiply,add and transpose of numpy module
print('\n addition\n',np.add(m1,m2))
print('\ntranspose\n',m1.transpose())
```

Output:

```
PS E:\books and pdfs\sem3 pdfs\python lab\week 11> python 11_2.py
array 1
 [[1 4 7]
 [2 5 8]]
array 2
 [[1 4 8]
 [2 3 6]]

Multiplication
 [[ 1 16 56]
 [ 4 15 48]]

 addition
 [[ 2  8 15]
 [ 4  8 14]]

transpose
 [[1 2]
 [4 5]
 [7 8]]
PS E:\books and pdfs\sem3 pdfs\python lab\week 11>
```