PYTHON LAB RECORD

WEEK1:

a)

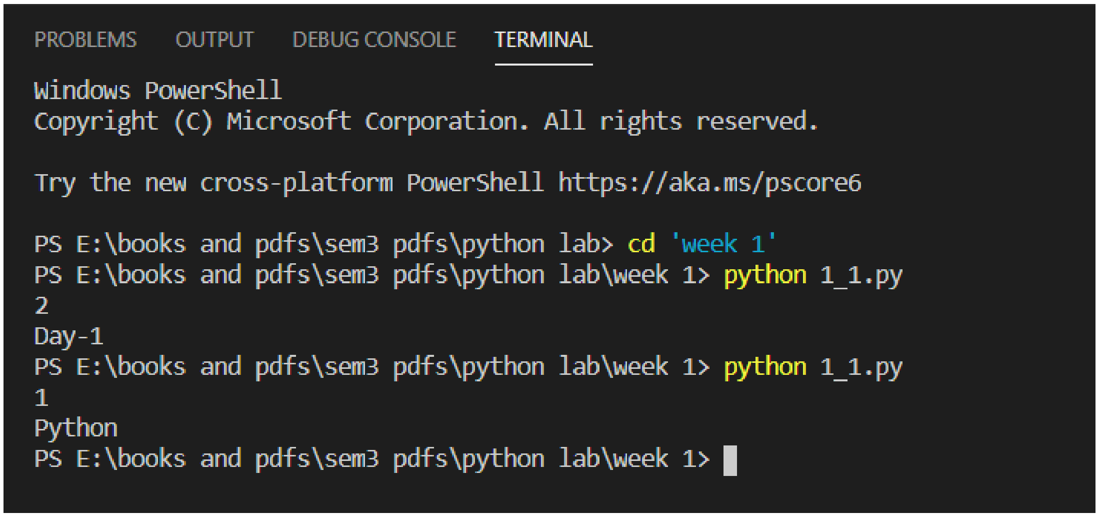
AIM:

Print the "Python" for 1, print "Day - 1" for 2. By changing the variable "look" for each statement.

CODE

```
i=int(input())
if i==1: look="Python"
elif i==2: look="Day-1"
print(look)
```

OUTPUT:



```
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> 
```

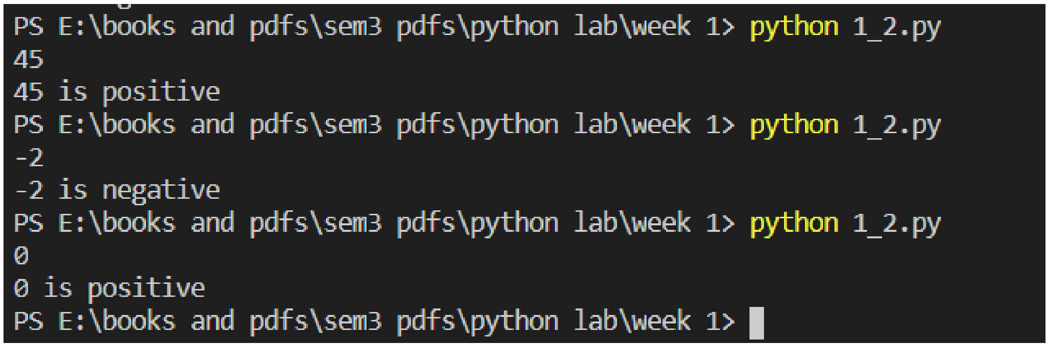
b)

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

CODE:

```
n=int((input()))
print(n,"is positive") if n>=0 else print(n,"is negative")
```

Output:



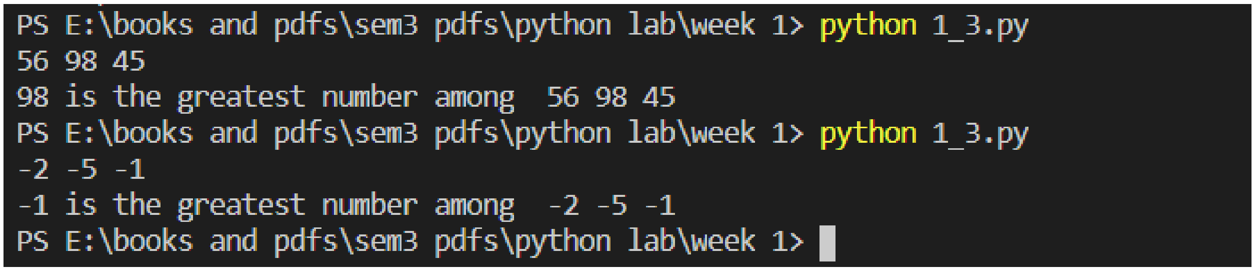
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_2.py
45
45 is positive
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_2.py
-2
-2 is negative
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_2.py
0
0 is positive
PS E:\books and pdfs\sem3 pdfs\python lab\week 1>
```

c)

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

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



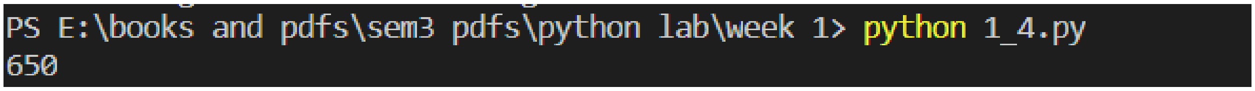
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_3.py
56 98 45
98 is the greatest number among  56 98 45
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_3.py
-2 -5 -1
-1 is the greatest number among  -2 -5 -1
PS E:\books and pdfs\sem3 pdfs\python lab\week 1>
```

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
for i in range(1,50):
    if i%2==0:
c+=i
print(c)
```

Output:



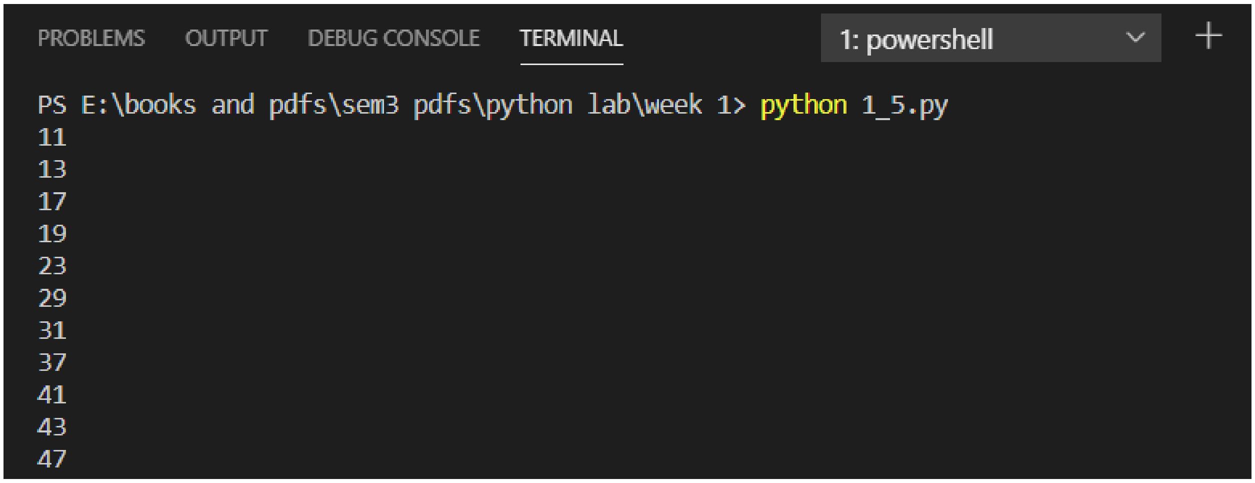
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 1> python 1_4.py
650
```

e)

AIM: Write a program to display all prime numbers within an interval of 20 and 50.

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

for i in range(10,51):
    if prime(i)==1:
print(i)
```

OUTPUT:



```
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
```

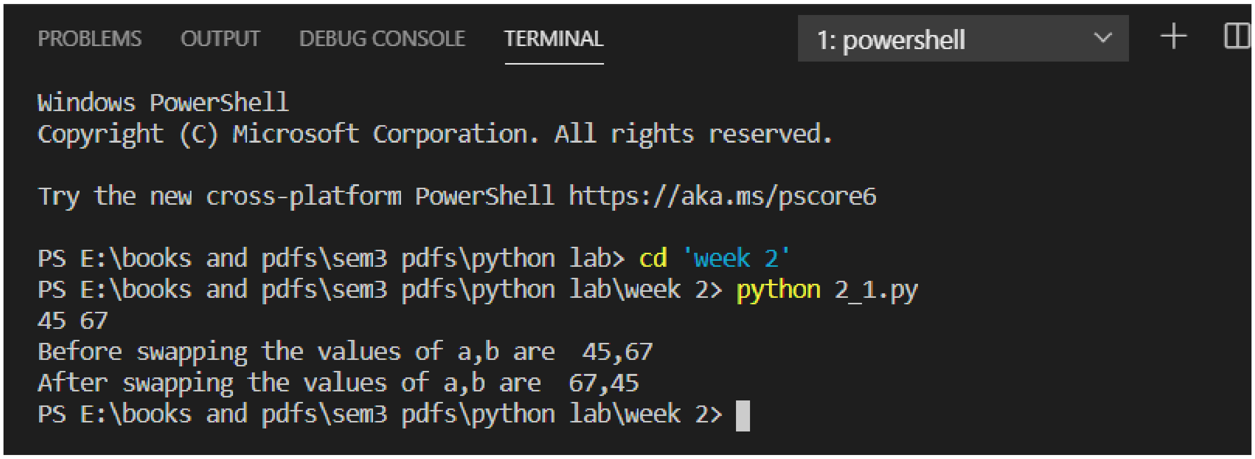
WEEK 2

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

Code:

```
a,b=input().split()
print("Before swapping the values of a,b are ",a+','+b)
a,b=b,a
print("After swapping the values of a,b are ",a+','+b)
```

Output:



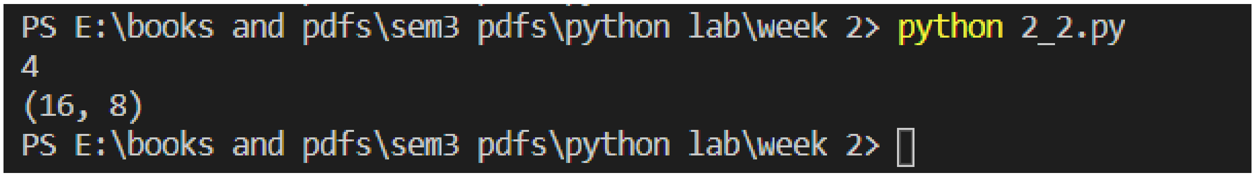
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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:

```
def func(n):
    return n*n,n+n

n=int(input())
print(func(n))
```

OUTPUT:

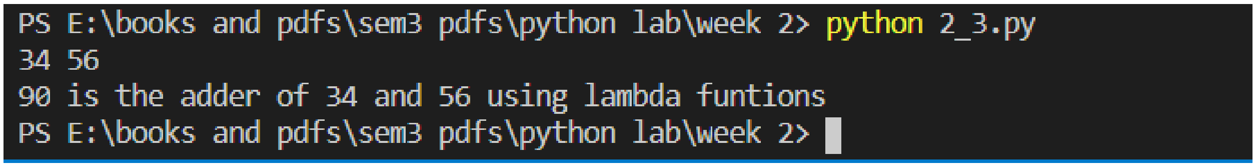


```
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> python 2_2.py
4
(16, 8)
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> 
```

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

CODE:

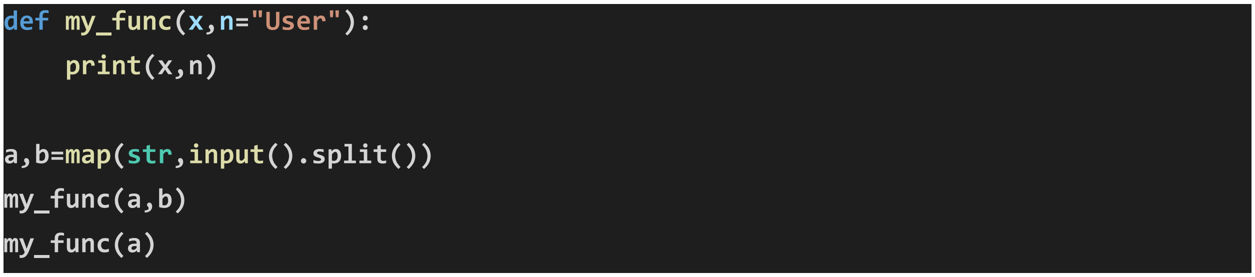
```
adder=lambda n,m:n+m
n,m=map(int,input().split())
print(adder(n,m) ,"is the adder of",n,"and",m,"using lambda funtions")
```



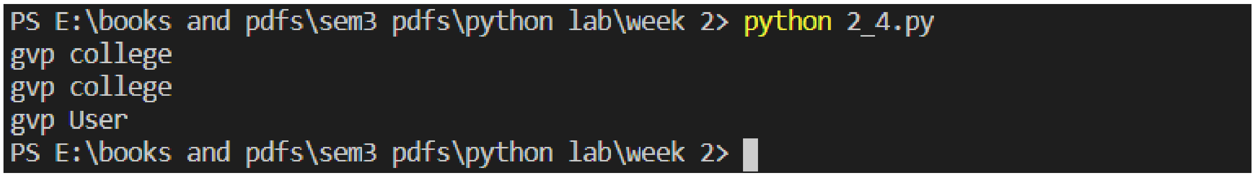
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> python 2_3.py
34 56
90 is the adder of 34 and 56 using lambda funtions
PS E:\books and pdfs\sem3 pdfs\python lab\week 2> 
```

d)

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



```
def my_func(x,n="User"):
    print(x,n)

a,b=map(str,input().split())
my_func(a,b)
my_func(a)
```



```
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:

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

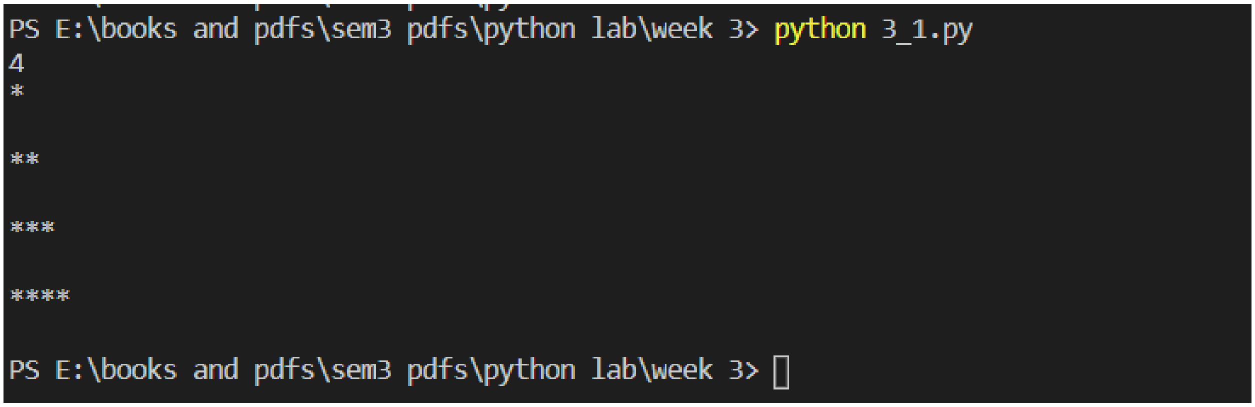
**

***

****

Code:

```
n=int(input())
for i in range(1,n+1):
    print(i* '*','\n')
```

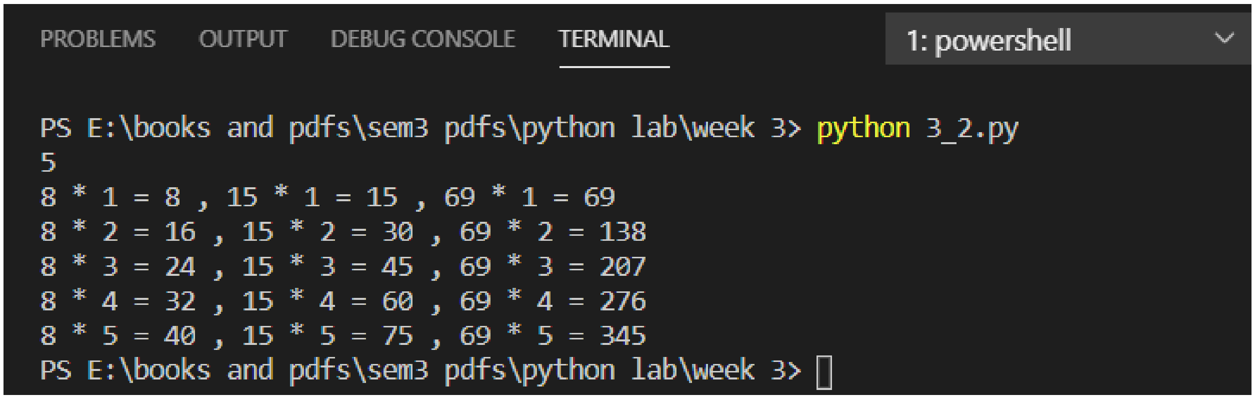


```
PS E:\books and pdfs\sem3 pdfs\python lab\week 3> python 3_1.py
4
*

**

***

****

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())
for i in range(1,n+1):
    print(8,'*',i,'=',8*i,',',15,'*',i,'=',15*i,',',69,'*',i,'=',69*i)
```

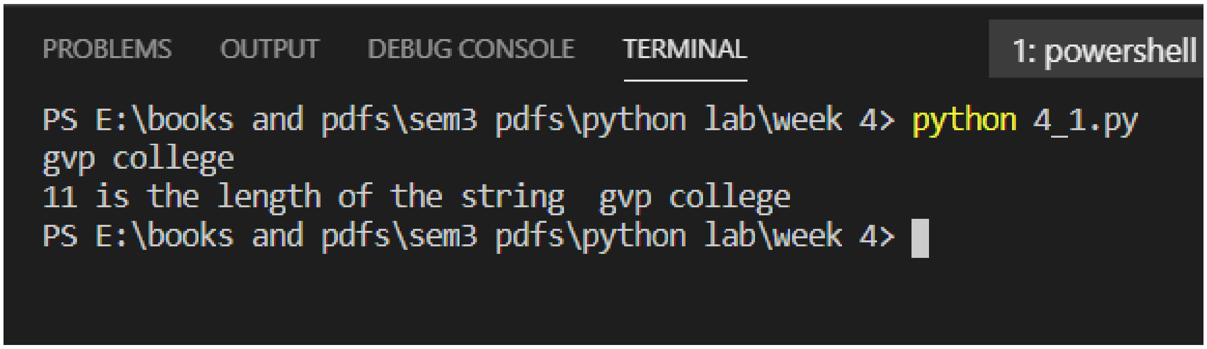


```
PS E:\books and pdfs\sem3 pdfs\python lab\week 3> python 3_2.py
5
8 * 1 = 8 , 15 * 1 = 15 , 69 * 1 = 69
8 * 2 = 16 , 15 * 2 = 30 , 69 * 2 = 138
8 * 3 = 24 , 15 * 3 = 45 , 69 * 3 = 207
8 * 4 = 32 , 15 * 4 = 60 , 69 * 4 = 276
8 * 5 = 40 , 15 * 5 = 75 , 69 * 5 = 345
PS E:\books and pdfs\sem3 pdfs\python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

WEEK 4:

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

```
str1=input()
c=0
for i in str1:
    c+=1
print(c,"is the length of the string ",str1)
```



```
PS E:\books and pdfs\sem3 pdfs\python lab\week 4> python 4_1.py
gvp college
11 is the length of the string  gvp college
PS E:\books and pdfs\sem3 pdfs\python lab\week 4>
```

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

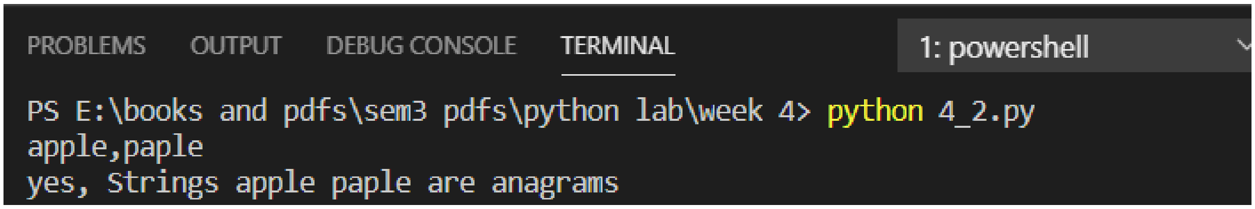
Code 1:

```
def fun(a,b):
    for i in a:
        if i not in b:
            return 0
    return 1
a,b=input().split(',')
print("yes, Strings "+a,b+ " are anagrams ") if fun(a,b) else print("no, 
Strings "+a,b +" are not anagrams ")
```

Code 2:

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

Output:



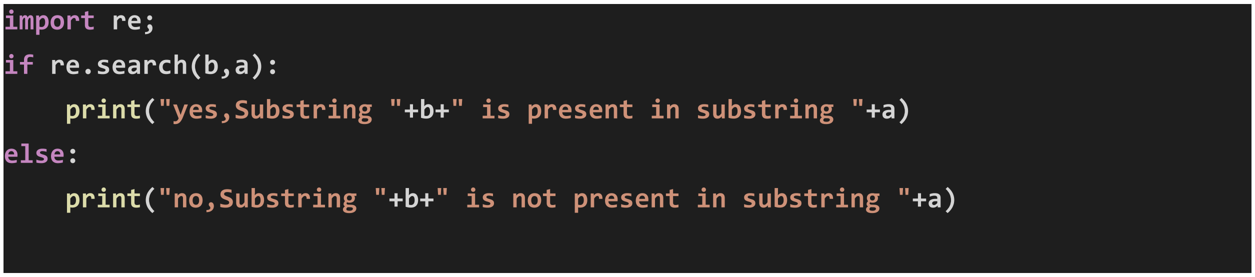
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 4> python 4_2.py
apple,paple
yes, Strings apple paple are anagrams
```

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

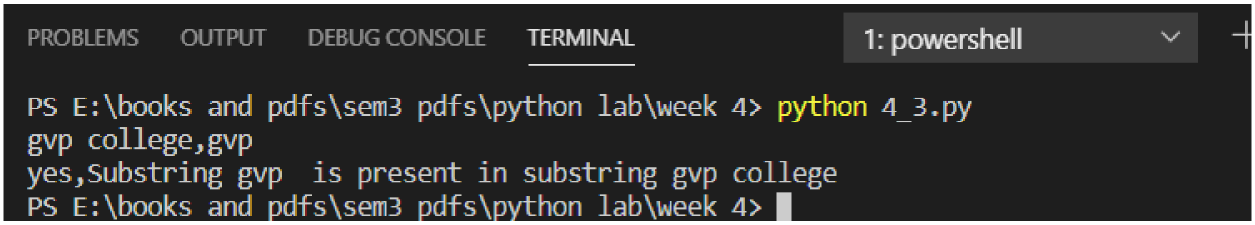
Code 1:

```
def fun(a,b):
    if b in a:
return 1
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:



```
import re;
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)
```



```
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> 
```

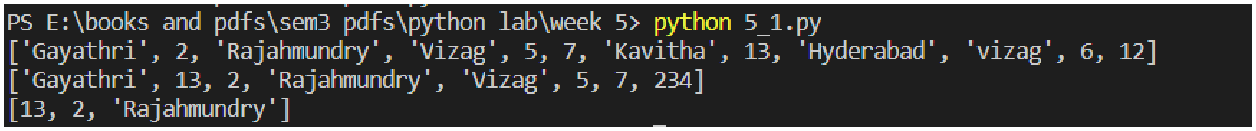
WEEK 5

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

Code:

```
li1=['Gayathri',2,'Rajahmundry','Vizag',5,7]
li2=['Kavitha',13,'Hyderabad','vizag',6,12]
print(li1+li2)
li1.append(234)
li1.insert(1,13)
print(li1)
print(li1[1:4])
```

Output:



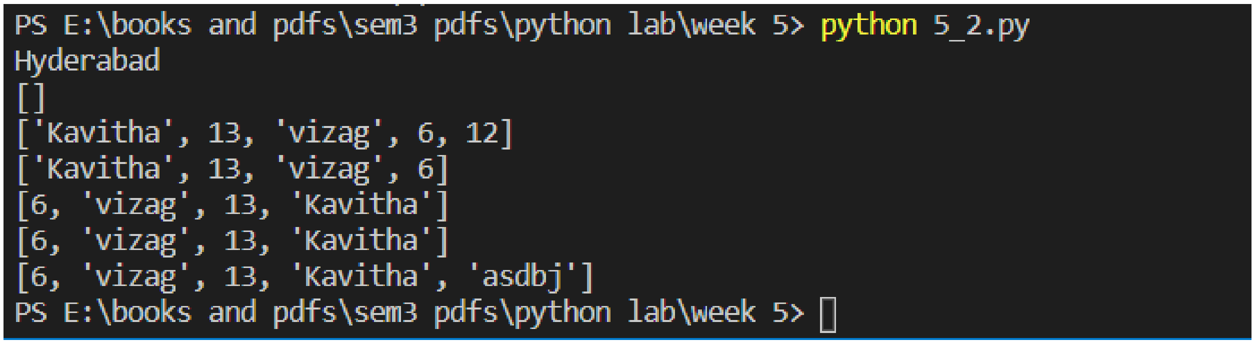
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> python 5_1.py
['Gayathri', 2, 'Rajahmundry', 'Vizag', 5, 7, 'Kavitha', 13, 'Hyderabad', 'vizag', 6, 12]
['Gayathri', 13, 2, 'Rajahmundry', 'Vizag', 5, 7, 234]
[13, 2, 'Rajahmundry']
```

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

Code:

```
li1=['Gayathri',2,'Rajahmundry','Vizag',5,7]
li2=['Kavitha',13,'Hyderabad','vizag',6,12]
print(li2.pop(2))
li1.clear()
print(li1)
print(li2)
li2.remove(12)
print(li2)
li2.reverse()
print(li2)
li1=li2.copy()
print(li1)
li1.append('asdbj')
print(li1)
del li1
```

Output:



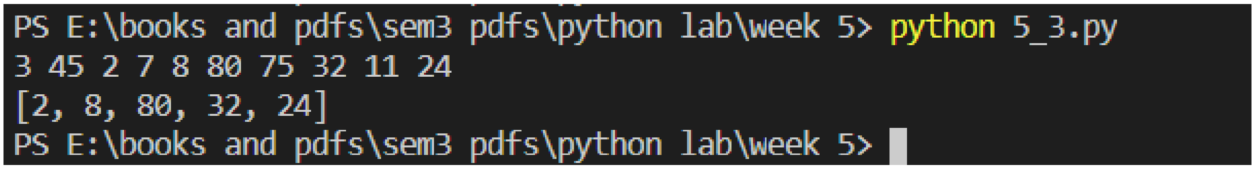
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> python 5_2.py
Hyderabad
[]
['Kavitha', 13, 'vizag', 6, 12]
['Kavitha', 13, 'vizag', 6]
[6, 'vizag', 13, 'Kavitha']
[6, 'vizag', 13, 'Kavitha']
[6, 'vizag', 13, 'Kavitha', 'asdbj']
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> 
```

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()))
print([i for i in li if i%2==0])
```

Output:

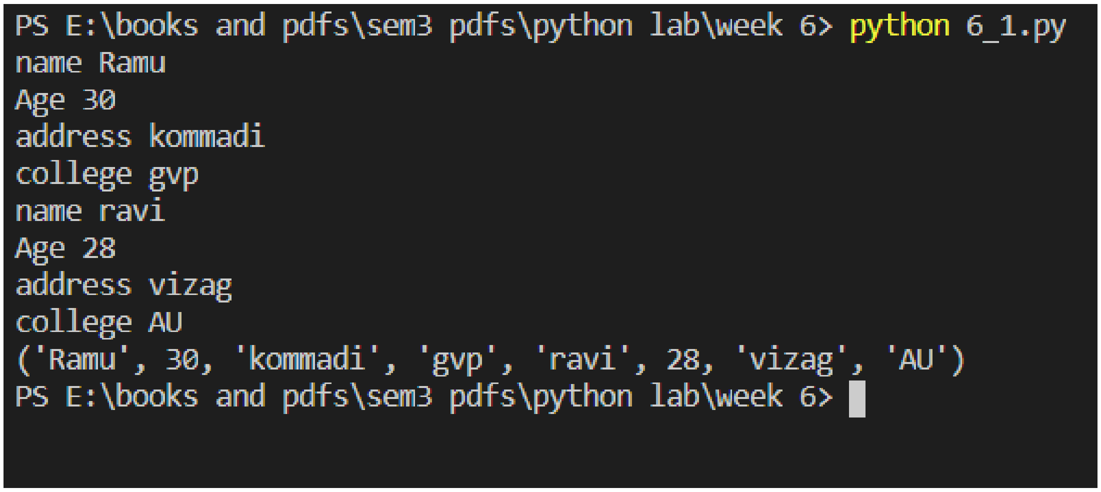


```
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> python 5_3.py
3 45 2 7 8 80 75 32 11 24
[2, 8, 80, 32, 24]
PS E:\books and pdfs\sem3 pdfs\python lab\week 5> 
```

WEEK 6:

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.**

```
t1=tuple((input("name "),int(input("Age ")),input("address "),input("college
")))
t2=tuple((input("name "),int(input("Age ")),input("address "),input("college
")))
#t1=t1+t2
#print(t1)
print(t1+t2)
```



```
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)]

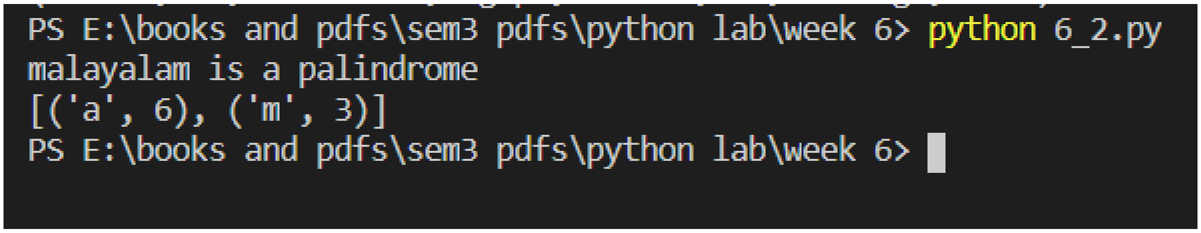
Code 1:

```
a=input()
li=[(i,a.count(i)) for i in a]
li=list(dict.fromkeys(li))
li= sorted(li, key=lambda tup: tup[1],reverse=True)
print(li[0:2])
```

Code 2:

```
from collections import Counter
counts = Counter(input())
print(counts.most_common(2))
```

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>
```

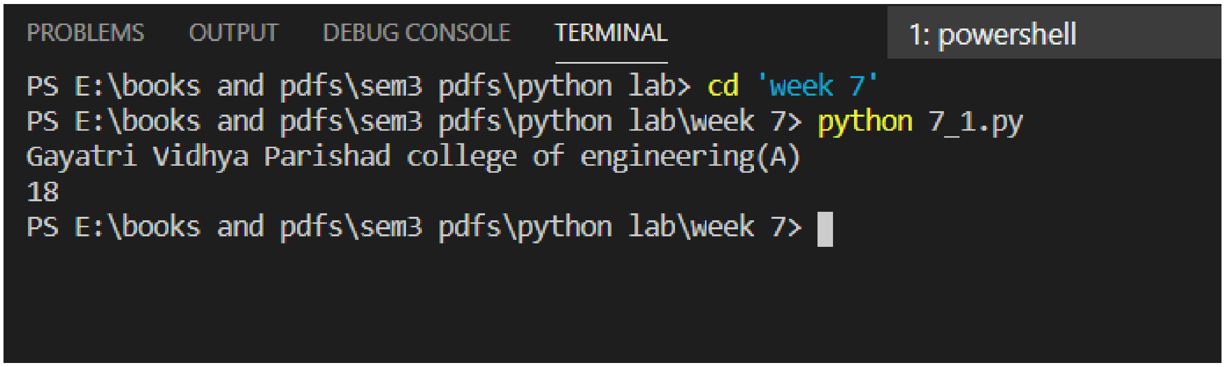
WEEK 7:

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

Code 1:

```
a=input()
b=set("aeiouAEIOU")
print(sum([1 for i in a if i in b]))
```

Output:



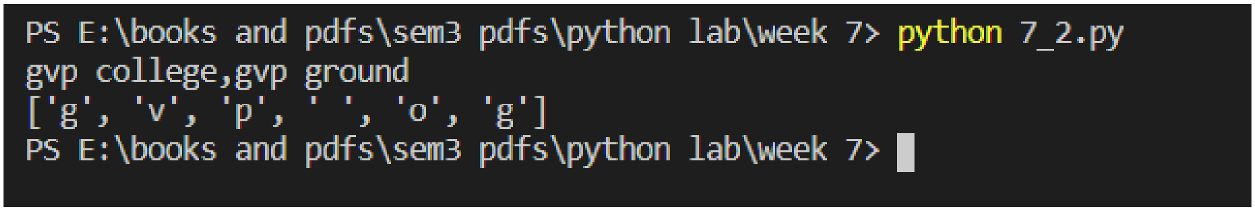
```
PS E:\books and pdfs\sem3 pdfs\python lab> cd 'week 7'
PS E:\books and pdfs\sem3 pdfs\python lab\week 7> python 7_1.py
Gayatri Vidhya Parishad college of engineering(A)
18
PS E:\books and pdfs\sem3 pdfs\python lab\week 7>
```

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

Code:

```
a,b=set(input().split(','))
print([i for i in a if i in b])
```

Output:



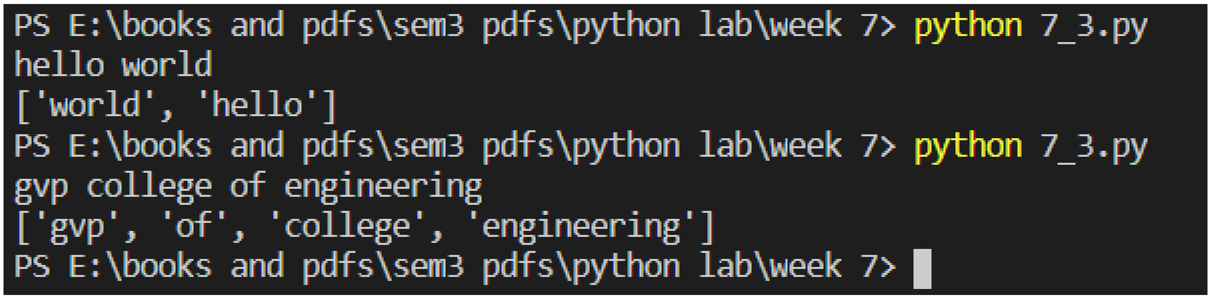
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 7> python 7_2.py
gvp college,gvp ground
['g', 'v', 'p', ' ', 'o', 'g']
PS E:\books and pdfs\sem3 pdfs\python lab\week 7>
```

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

Code:

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

n=input().split()
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:

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

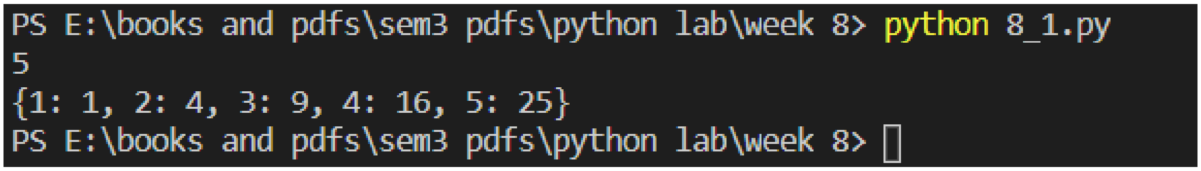
Code 1:

```
n=int(input())
print({i:i*i for i in range(n)})
```

Code 2:

```
res=dict()
for i in range(n):
    res[i]=i*i
print(res)
```

Output:



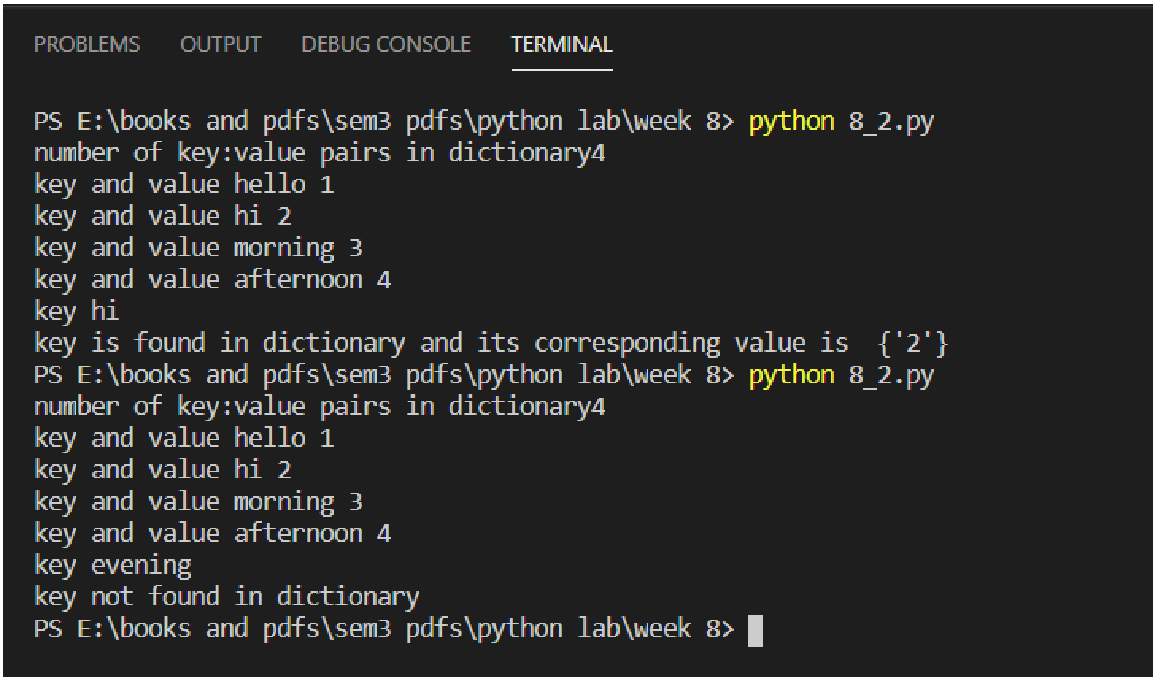
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_1.py
5
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> ▯
```

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

Code:

```
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 ")
l={d[i] for i in d if i==k}
if l:print("key is found in dictionary and its corresponding value is ",l)
else:print("key not found in dictionary")
```

Output:



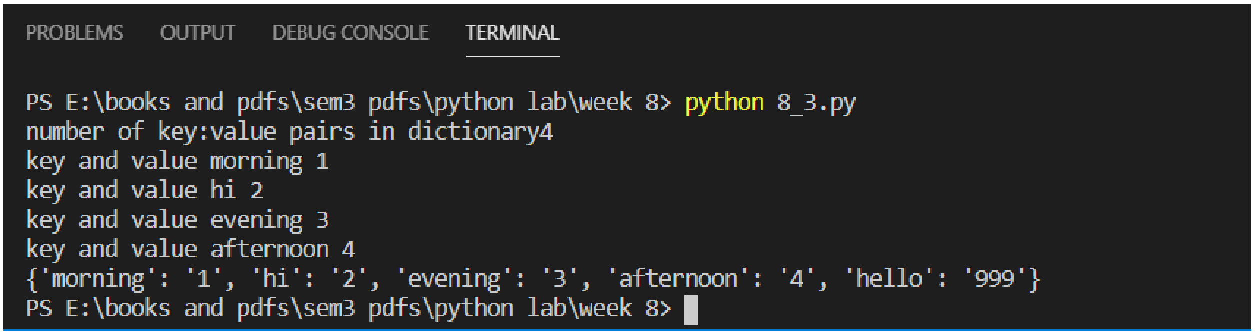
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_2.py
number of key:value pairs in dictionary4
key and value hello 1
key and value hi 2
key and value morning 3
key and value afternoon 4
key hi
key is found in dictionary and its corresponding value is  {'2'}
PS E:\books and pdfs\sem3 pdfs\python 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:

```
n=int(input("number of key:value pairs in dictionary"))
d=dict(input("key and value ").split() for i in range(n))
d.update({"hello":"999"})
print(d)
```

Output:



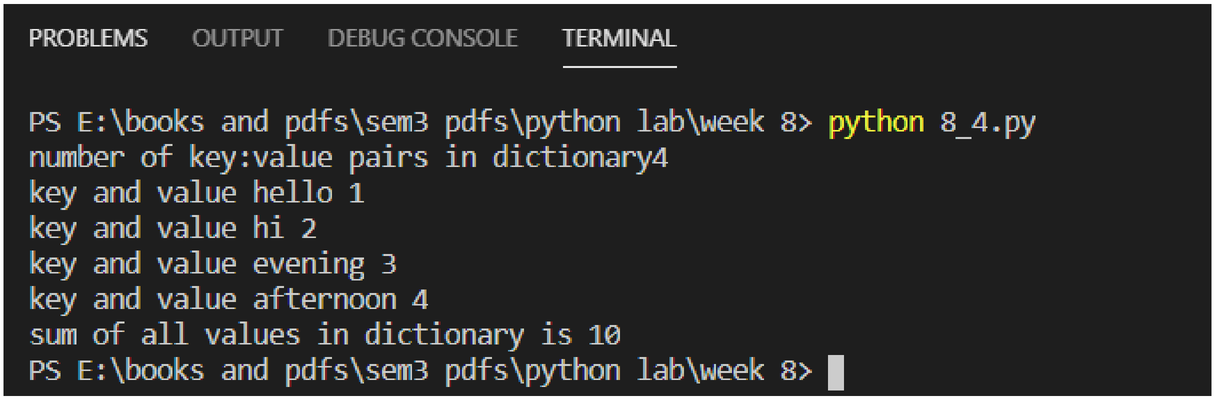
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 8> python 8_3.py
number of key:value pairs in dictionary4
key and value morning 1
key and value hi 2
key and value evening 3
key and value afternoon 4
{'morning': '1', 'hi': '2', 'evening': '3', 'afternoon': '4', 'hello': '999'}
PS E:\books and pdfs\sem3 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))
print("sum of all values in dictionary is",sum([int(d[i]) for i in d]))
```

Output:



```
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>
```

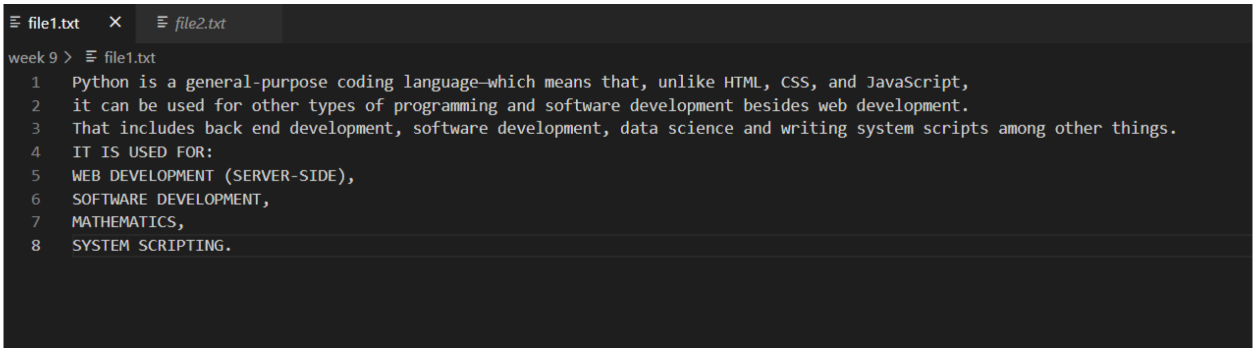
WEEK 9:

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)**

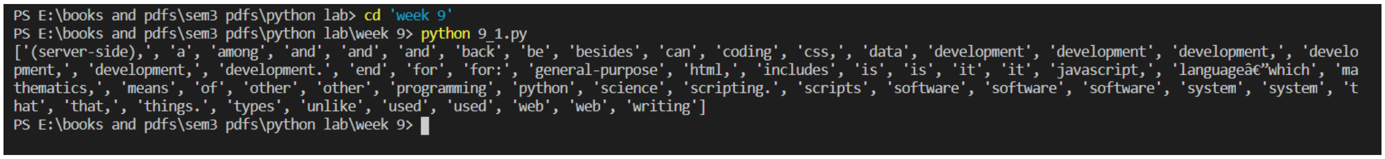
Code:

```
f1 = open("file1.txt",'r')
l = f1.read().split("\n")
lt = []
for i in l:
    for j in i.split():
        lt.append(j.lower())
lt.sort()
print(lt)
f2 = open("output.txt", "w")
for i in lt:
    f2.write(str(i)+"\n")
f1.close()
f2.close()
```

file1.txt:

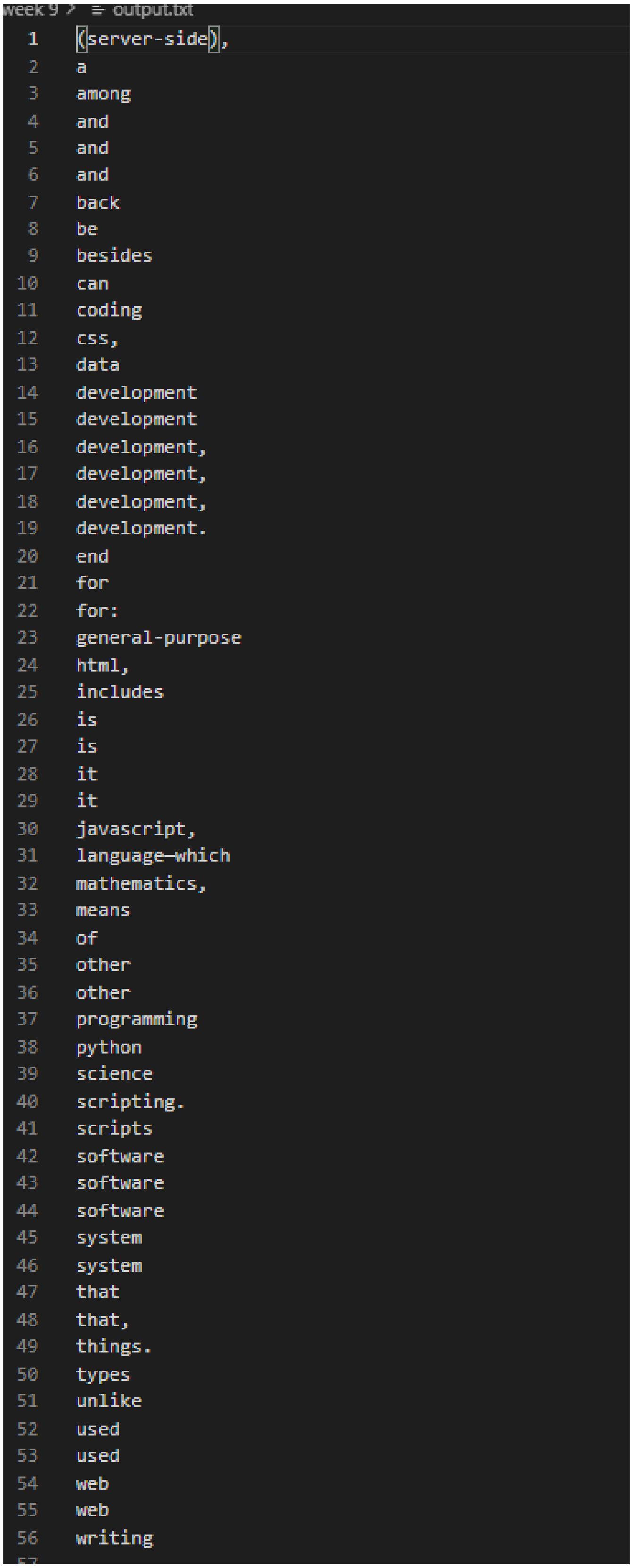


```
Python is a general-purpose coding language—which means that, unlike HTML, CSS, and JavaScript,
it can be used for other types of programming and software development besides web development.
That includes back end development, software development, data science and writing system scripts among other things.
IT IS USED FOR:
WEB DEVELOPMENT (SERVER-SIDE),
SOFTWARE DEVELOPMENT,
MATHEMATICS,
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:



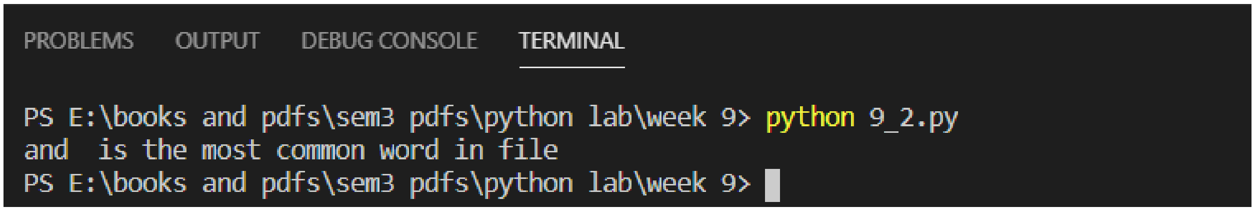
```
(server-side),
a
among
and
and
and
back
be
besides
can
coding
css,
data
development
development
development,
development,
development,
development.
end
for
for:
general-purpose
html,
includes
is
is
it
it
javascript,
language—which
mathematics,
means
of
other
other
programming
python
science
scripting.
scripts
software
software
software
system
system
that
that,
things.
types
unlike
used
used
web
web
writing
```

b) **Aim: Write a program to find the most frequent words in a text.(read from a text file)**

Code:

```
from collections import Counter
li=[]
f=open("file1.txt","r")
for i in f:
    for j in i.split():
        if j!='\n' or j!='\t':
            li.append(j)

mo=Counter(li)
print(mo.most_common(1)[0][0]," is the most common word in file")
```

Output:



```
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> 
```

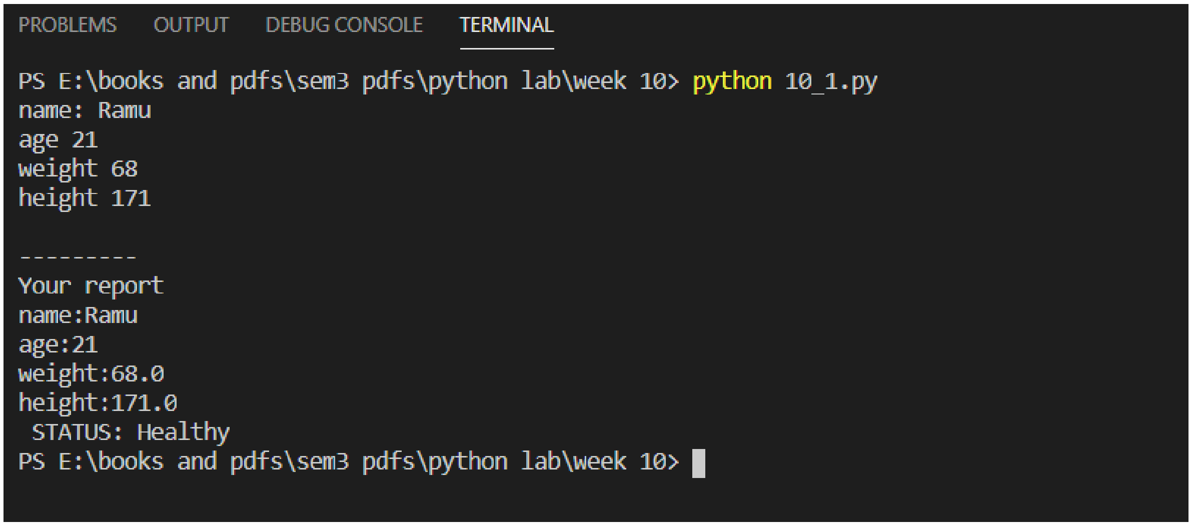
WEEK 10:

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:

```
class Person:
    def __init__(self,name,age,weight,height):
        self.name=name
        self.age=age
        self.weight=weight
        self.height=height
    def get_bmi_result(self):
        h=self.height/100
        bmi=self.weight/(h**2)
        if bmi<=18.5:
            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:
            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:
            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()
print(Person.get_bmi_result(p1))
```

Output:



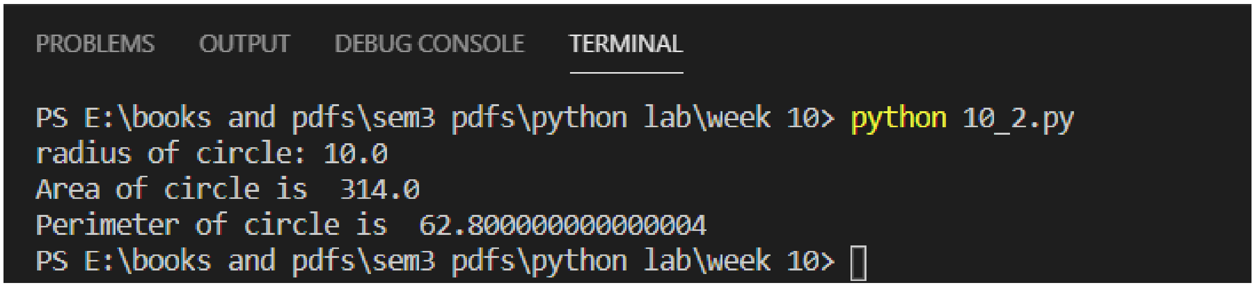
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 10> python 10_1.py
name: Ramu
age 21
weight 68
height 171

---------
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:

```
class circle:
    def __init__(self,radius):
        self.radius=radius
    def perimeter(self):
        return 2*3.14*self.radius

    def area(self):
        return 3.14*self.radius*self.radius



c1=circle(float(input("radius of circle: ")))
c1.area()
print("Area of circle is ",circle.area(c1))
c1.perimeter()
print("Perimeter of circle is ",circle.perimeter(c1))
```

Output:



```
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> 
```

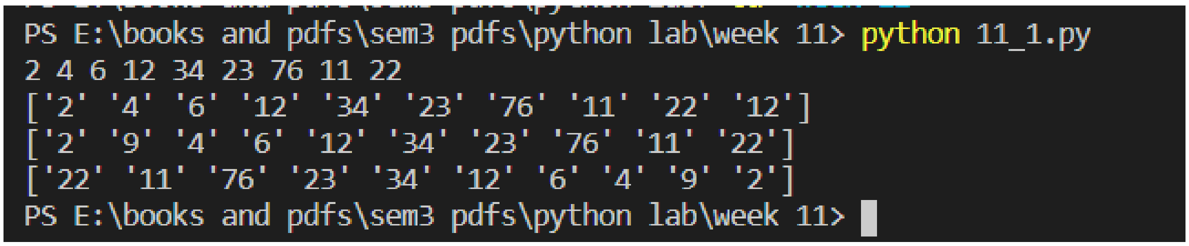
WEEK 11:

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

Code:

```
import numpy as np
arr=np.array(input().split())
print(np.append(arr,12))
arr=np.insert(arr,1,9)
print(arr)
print(np.flip(arr))
```

Output:



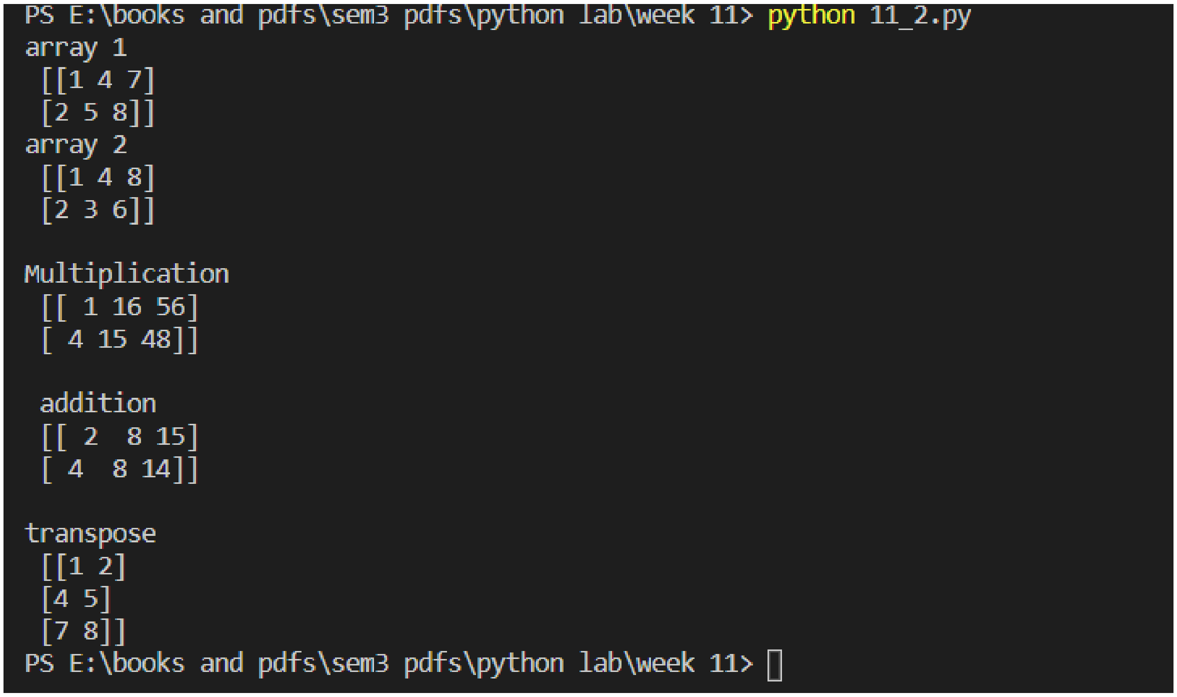
```
PS E:\books and pdfs\sem3 pdfs\python lab\week 11> python 11_1.py
2 4 6 12 34 23 76 11 22
['2' '4' '6' '12' '34' '23' '76' '11' '22' '12']
['2' '9' '4' '6' '12' '34' '23' '76' '11' '22']
['22' '11' '76' '23' '34' '12' '6' '4' '9' '2']
PS E:\books and pdfs\sem3 pdfs\python lab\week 11> 
```

b) **Aim: Write a program to add, transpose and multiply two matrices.**

Code:

```
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))
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> 
```