**PYTHON LAB RECORD**

**WEEK1:**

**a)**

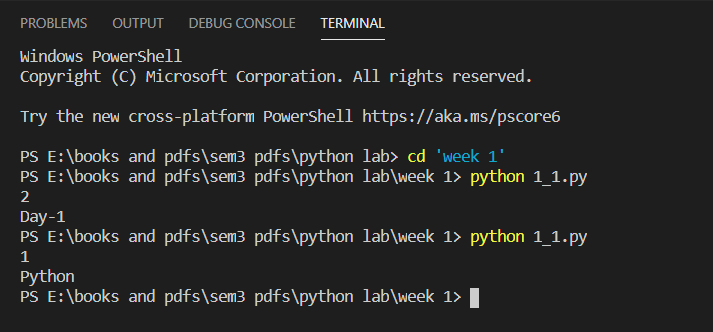
**AIM:**

**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:**

****

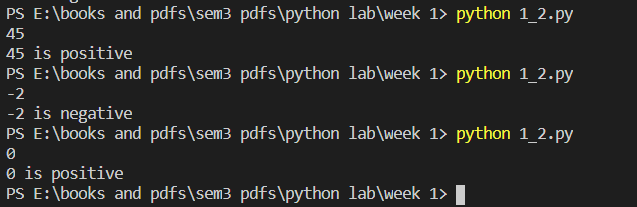
**b)**

**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:**

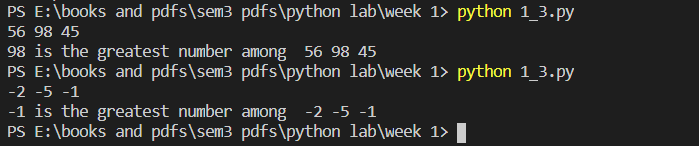
****

**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:**

****

**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:**

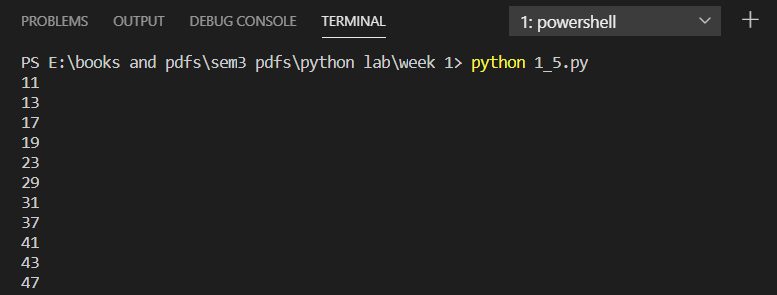
****

**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:**

****

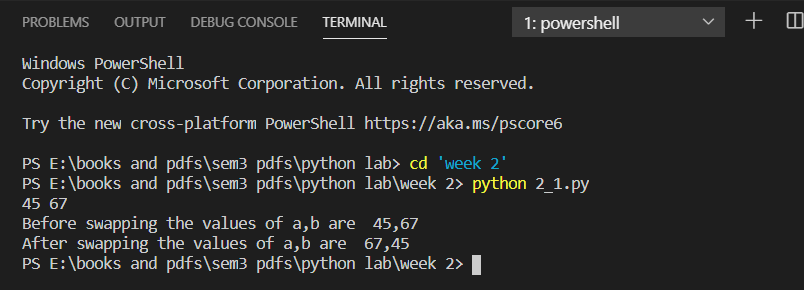
**WEEK 2**

**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:**

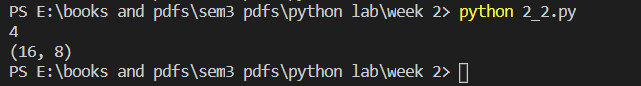
****

**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))** |

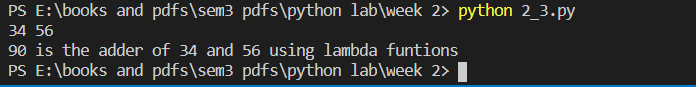
**OUTPUT:**

****

**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")** |

****

**d)**

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

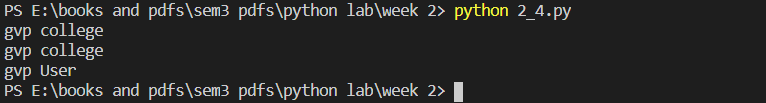
**def my\_func(x,n="User"):**

**print(x,n)**

**a,b=map(str,input().split())**

**my\_func(a,b)**

**my\_func(a)**

****

**WEEK 3:**

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

**\***

**\*\***

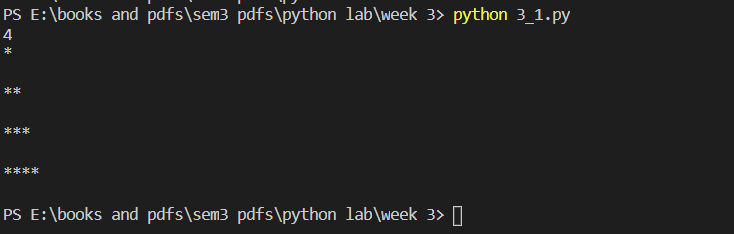
**\*\*\***

**\*\*\*\***

**Code:**

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

|  |
| --- |
|  |

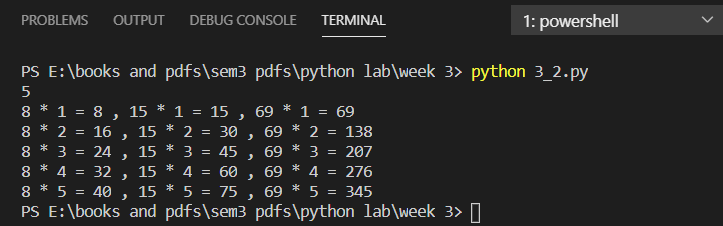
****

**b)**

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

**Code:**

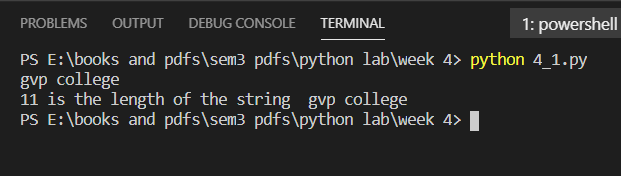
|  |
| --- |
| **n=int(input()) for i in range(1,n+1):  print(8,'\*',i,'=',8\*i,',',15,'\*',i,'=',15\*i,',',69,'\*',i,'=',69\*i)** |

****

**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)** |

****

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

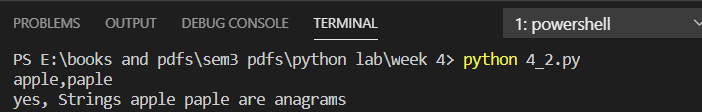
**Code 1:**

|  |
| --- |
| **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:**

****

**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:**

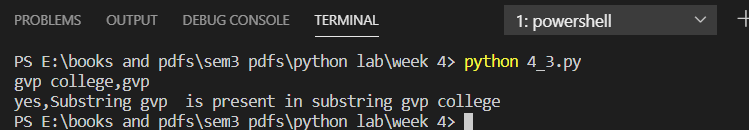
**import re;**

**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)**

****

**WEEK 5**

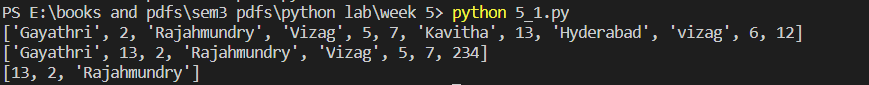
**a)Aim: Write a program to perform the given operations on a list:**

**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:**

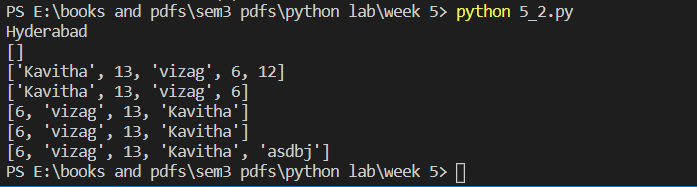
****

**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:**

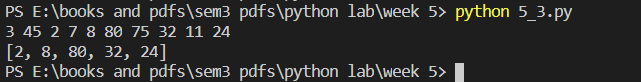
****

**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])** |

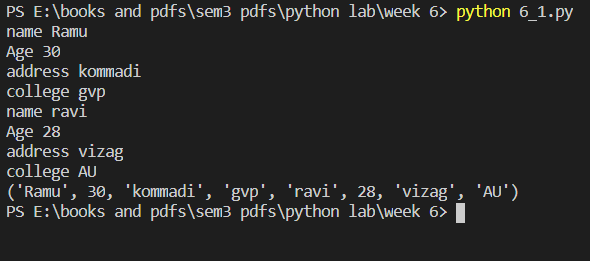
**Output:**

****

**WEEK 6:**

1. **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)** |

****

1. **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)]**

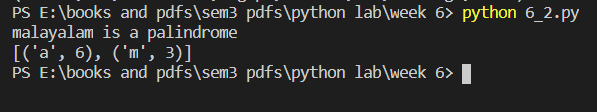
**Code 1:**

|  |
| --- |
| **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:**

****

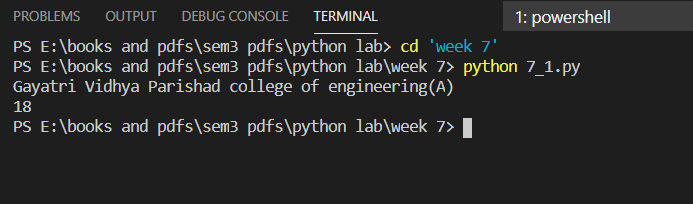
**WEEK 7:**

1. **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:**

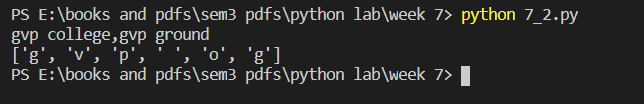
****

**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:**

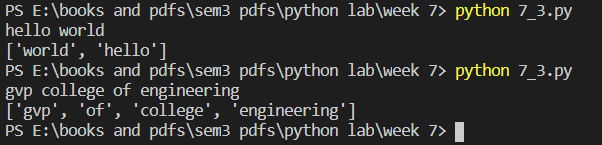
****

**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:**

****

**WEEK 8:**

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

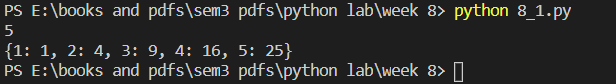
**Code 1:**

|  |
| --- |
| **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:**

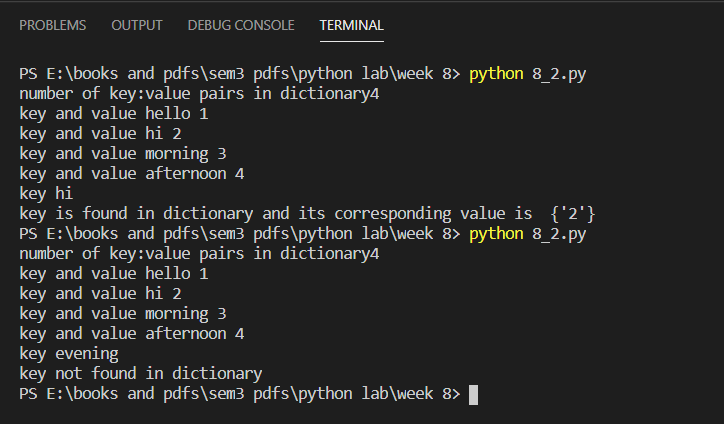
****

1. **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:**

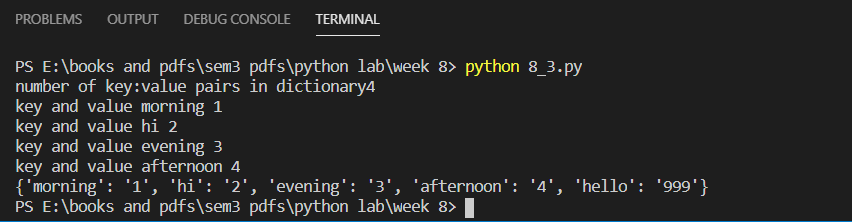
****

1. **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:**

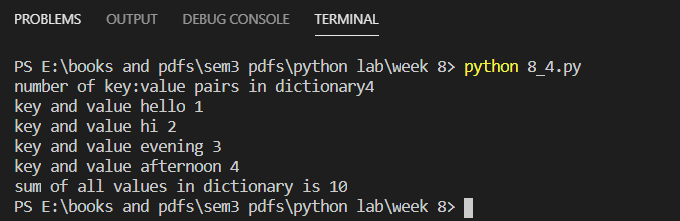
****

1. **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:**

****

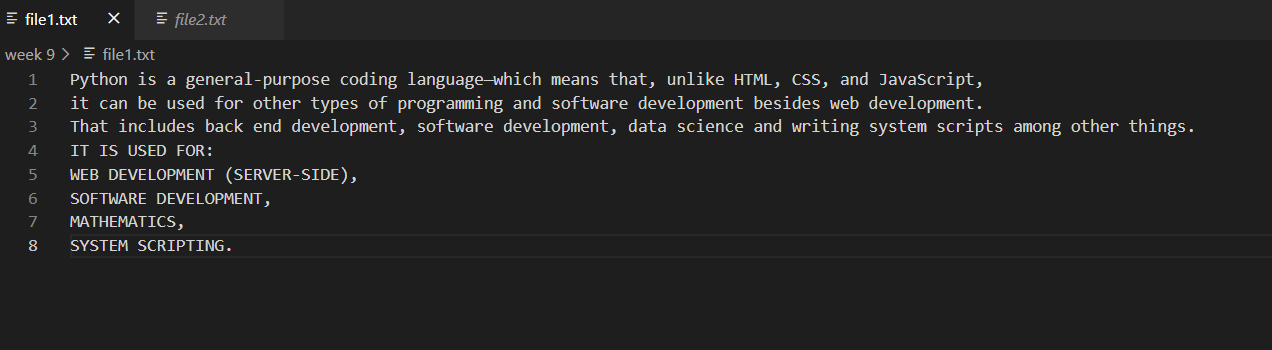
**WEEK 9:**

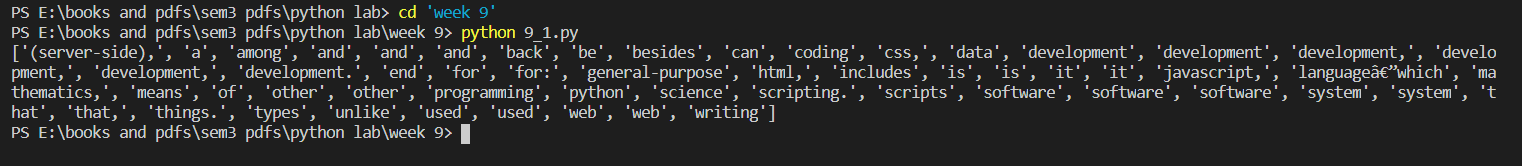
1. **Aim: a. Write a program to sort words in a file and put them in another file. The output file should have only lower case words, so any upper case words from source must be lowered. (Handle exceptions)**

**Code:**

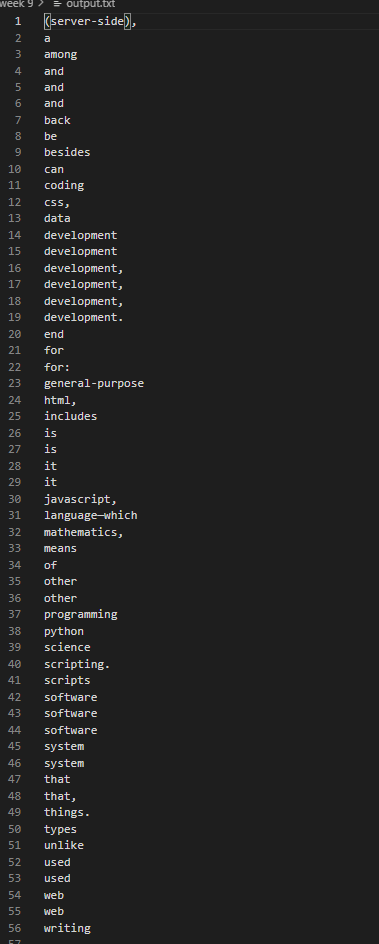
|  |
| --- |
| **f1 = open("file1.txt",'r') l = f1.read().split("\n") lt = [] for i in l:  for j in i.split():  lt.append(j.lower()) lt.sort() print(lt) f2 = open("output.txt", "w") for i in lt:  f2.write(str(i)+"\n") f1.close()  f2.close()** |

**file1.txt:**

****

****

**Output.txt:**

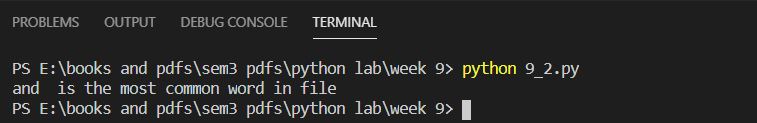
****

1. **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:**

****

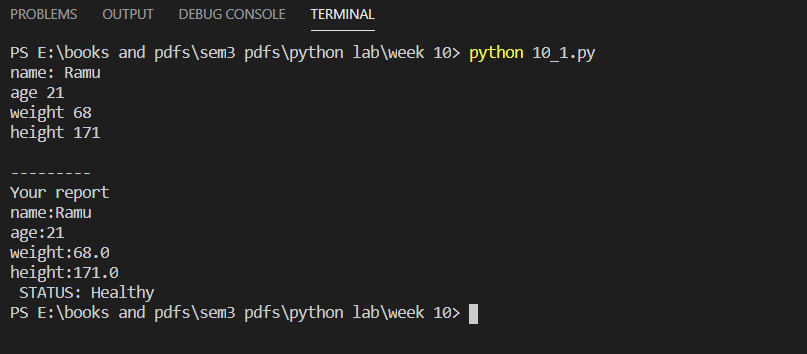
**WEEK 10:**

1. **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:**

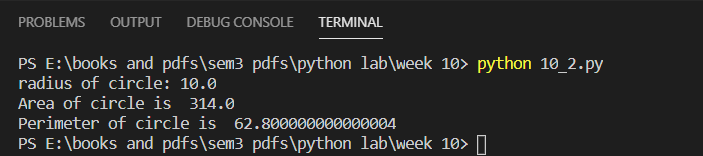
****

1. **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:**

****

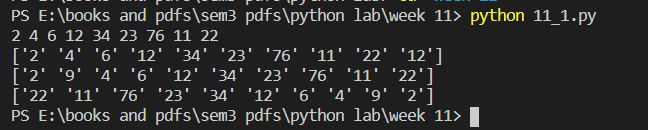
**WEEK 11:**

1. **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:**

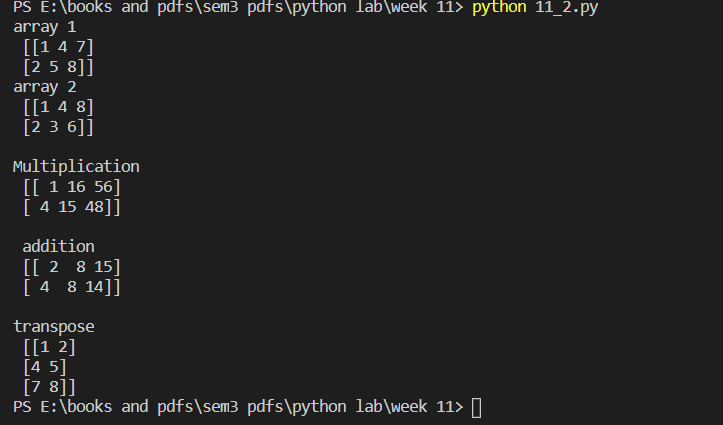
****

1. **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:**

****