**Batch Name:** Infosys FP5.0 Summer 2018

**Enrollment No:** R171217044

**SAPID:** 500060722

**Name:** PRAJJAWAL BANATI

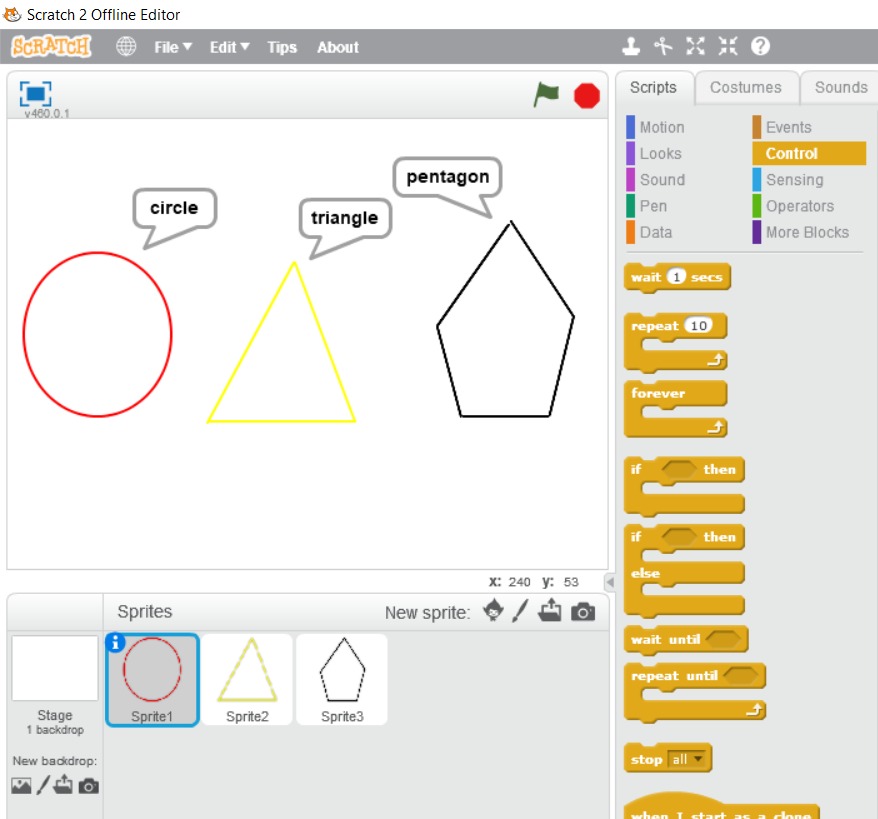
**Sem:** SEM-II

**Branch:** CSE-DEVOPS

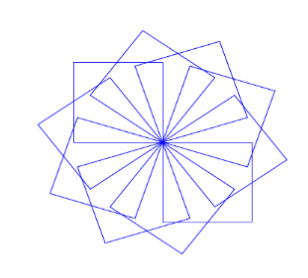
**ASSIGNMENT 1**

1. Take a text input from the user as one of the three shape names – "square", "triangle" or "pentagon". Based on the input, draw either a red square, yellow triangle or black pentagon.

Ans:



1. Create the following pattern using Scratch.

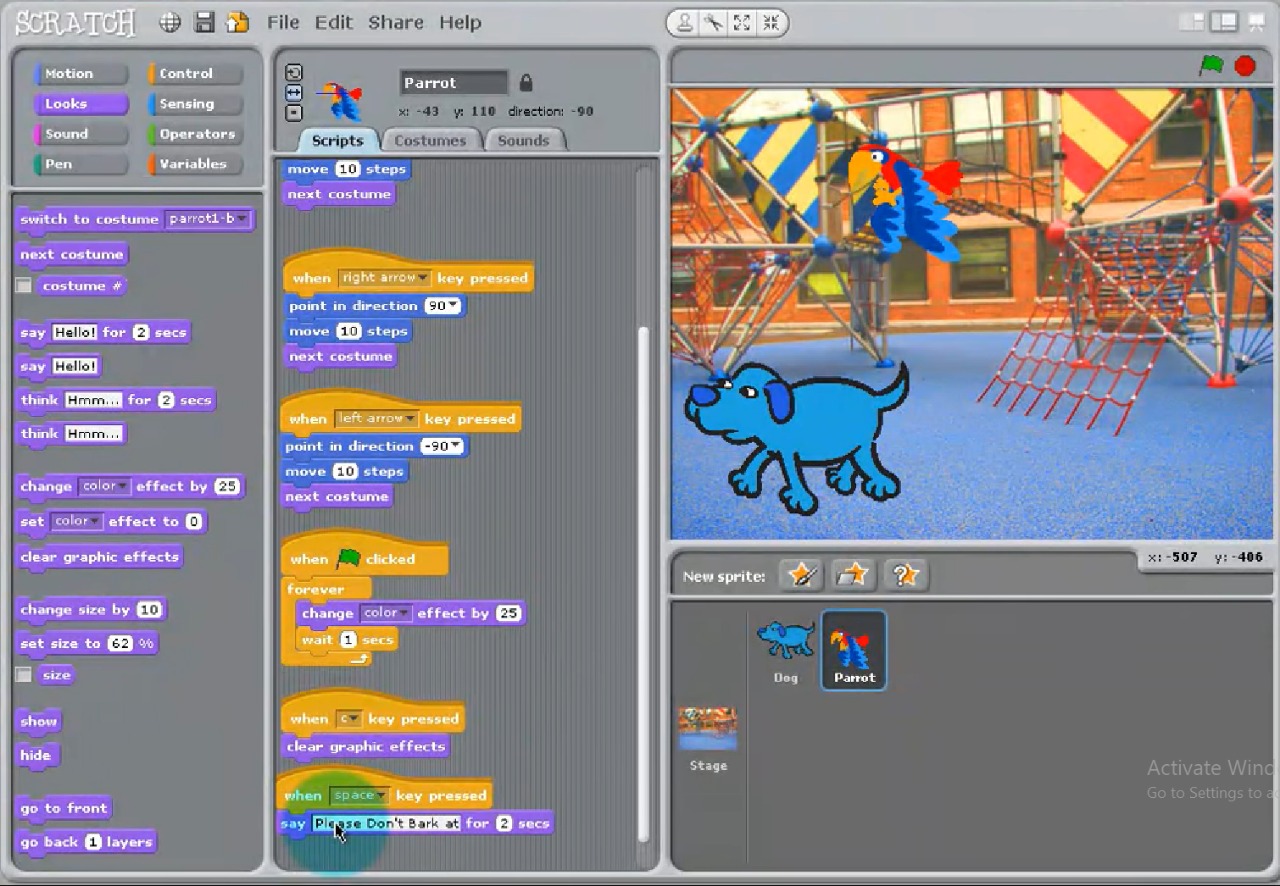


Ans:



1. Create or import two Sprites and use your imagination to make them do diﬀerent actions simultaneously. For example: "A bird is ﬂying and a dog is walking at the same time."

Ans:



**ASSIGNMENT 2**

1. Calculate the average of three numbers. If average is greater than or equal to 75, print "Pass", else print "Fail".

Ans:

**Print “Fail”**

**Print ”Pass”**

**d=(a+b+c)/3**

**INPUT a,b,c**

**Is d>75 ?**

YES

NO

1. Calculate and print the factorial of a number

Ans:

NO

YES

**C=1**

**C=C\*a  
a=a-1**

**INPUT a**

**Is a>0 ?**

**DISPLAY C**

1. Accept the lengths of three sides of a triangle as input from the user. Based on the input, print if the given triangle is "Equilateral", "Isosceles" or "Scalene".

Ans:

**PRINT “Isosceles Triangle”**

**PRINT “Equilateral Triangle”**

YES

YES

NO

NO

NO

**Is b=c ?**

YES

**Is a=b?**

**INPUT a,b,c**

**Is b=c ?**

**PRINT “Scalene Triangle”**

Accept the values of principal amount, rate of interest and number of years as an input from the user. Calculate and print the simple interest.

Ans:

**DISPLAY S.I.**

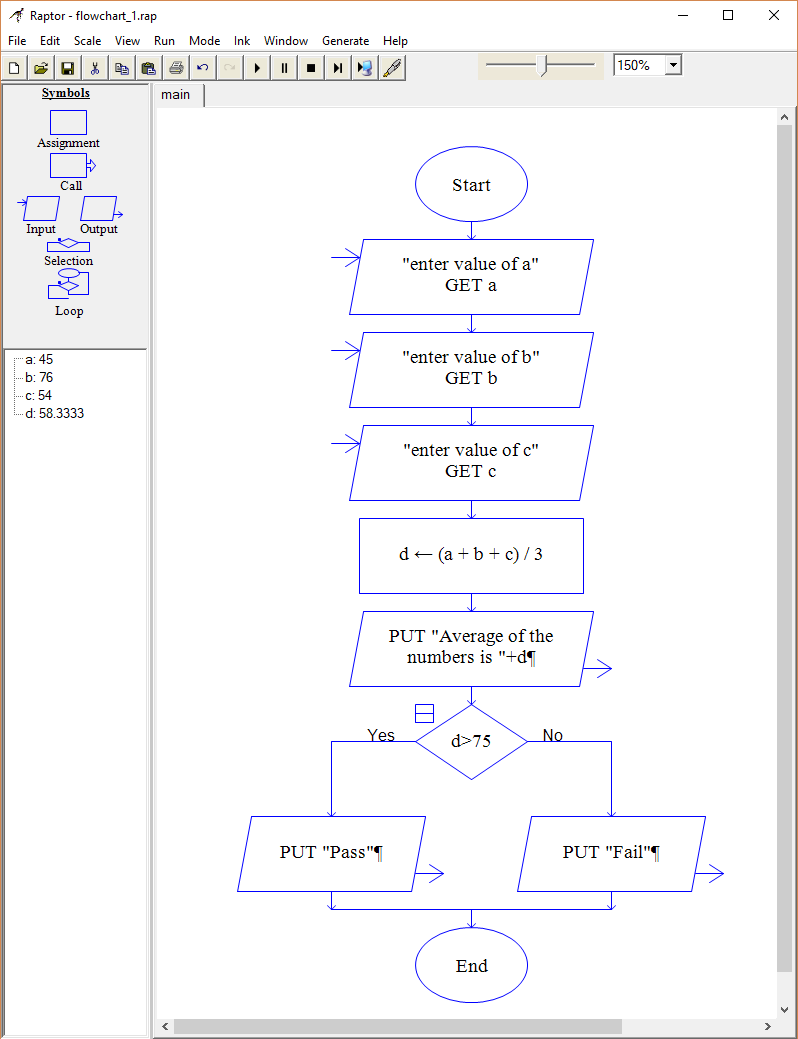
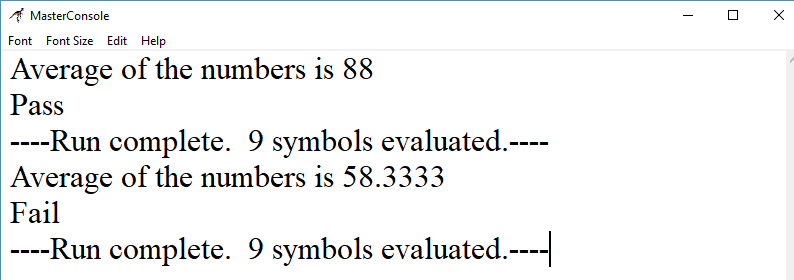
**S.I.=(P\*R\*T)/100**

**INPUT P,R,T**

**ASSIGNMENT 3**

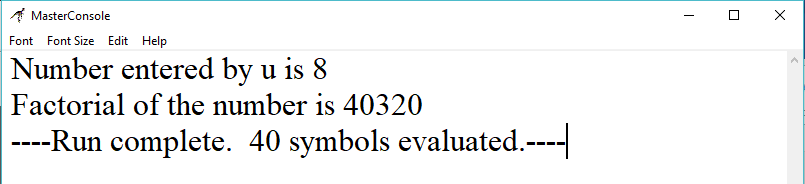
In previous section, you have created the ﬂowcharts for the following problems. Now, use Raptor tool to create and execute ﬂowcharts for these problems. Observe the output for diﬀerent set of inputs.

1. Calculate the average of three numbers. If average is greater than or equal to 75, print "Pass", else print "Fail".

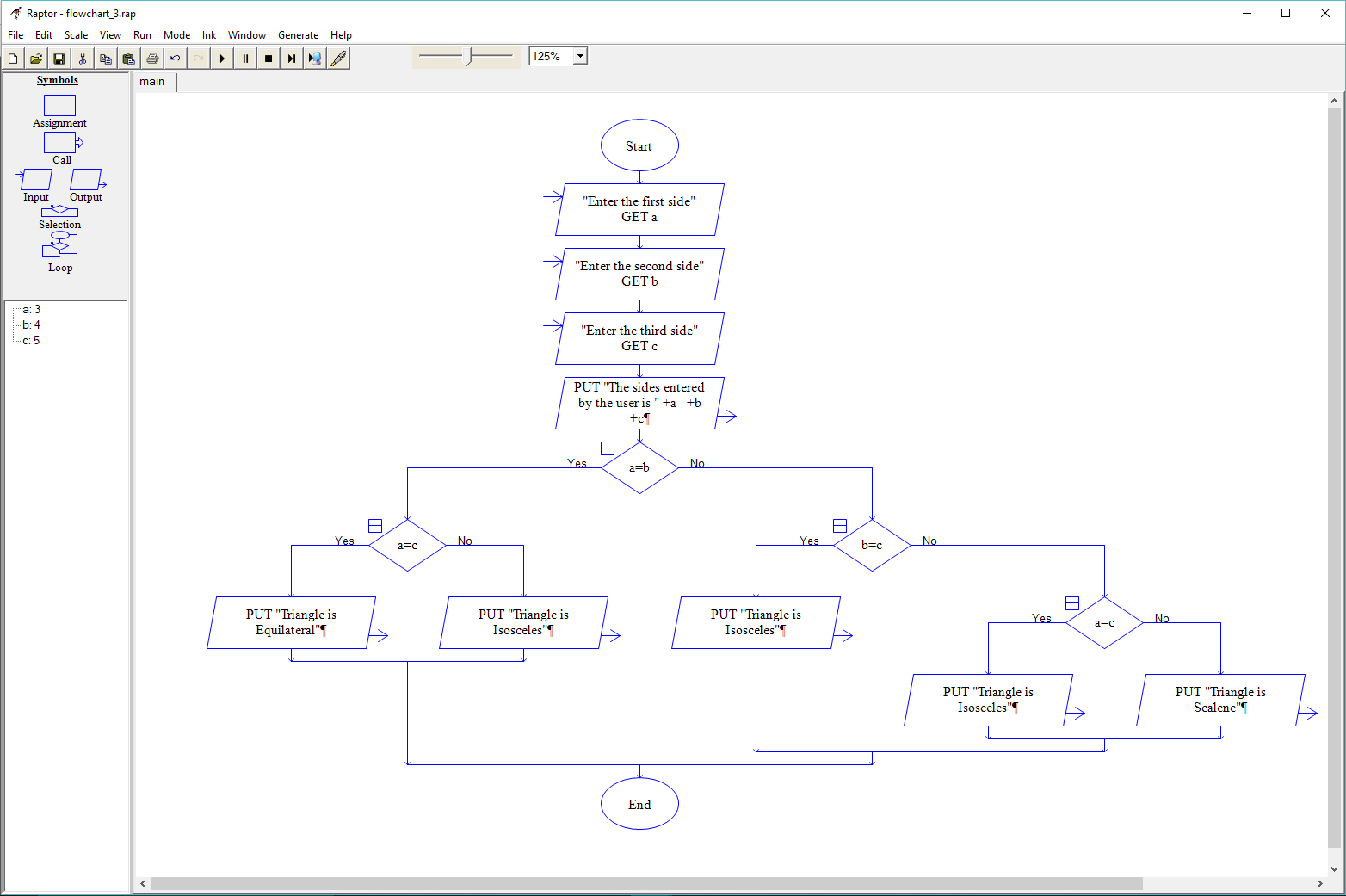
 

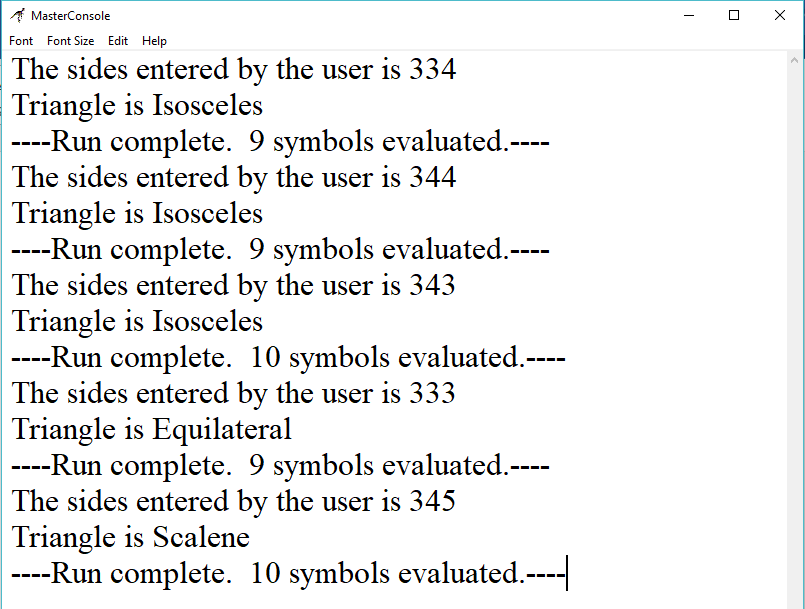
1. Calculate and print the factorial of a number



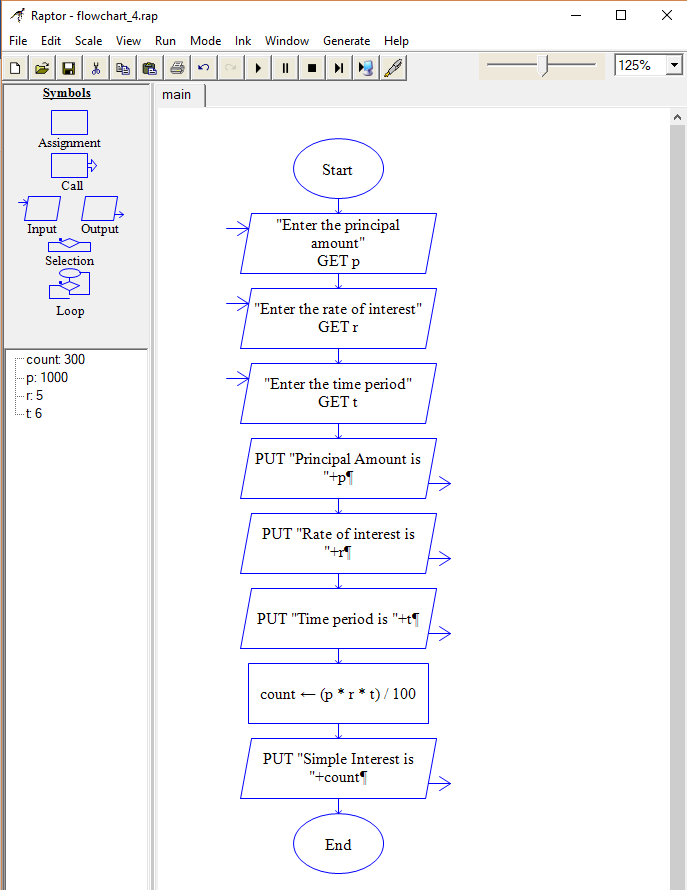


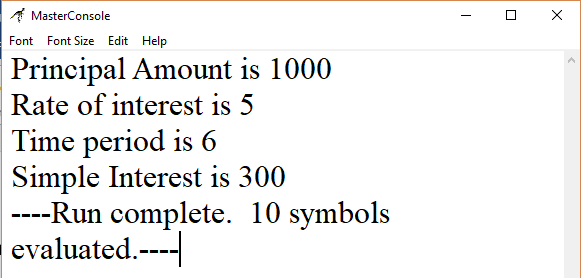
1. Accept the lengths of three sides of a triangle as input from the user. Based on the input, print if the given triangle is "Equilateral", "Isosceles" or "Scalene".





1. Accept the values of principal amount, rate of interest and number of years as an input from the user. Calculate and print the simple interest.





**ASSIGNMENT 4**

Q1: Write Pseudo Code:

1. To check whether a given number is even or odd.

Ans: **Step 1:** Start

**Step 2:** [Take Input] Read: N

**Step 3:** Check: If N%2 == 0 Then

Print : N is an Even Number.

Else

Print : N is an Odd Number.

**Step 4:** Exit

1. To ﬁnd factorial of a given number.

Ans: To calculate n!, given n

**Step 1**: Input integer number n

**Step 2**: If n< 0 , Output “error”, stop,

Else, Initialise Product to 1

**Step 3**: If n=0 or n=1, Output Product,Stop

Else, Initialise Multiplier to 2

**Step 4**: Redefine Product=Product\* Multiplier

**Step 5**: Increment Multiplier by 1

**Step 6**: If Multiplier is less than or equal to n , go to 6

Else, Output Product

1. To calculate ‘x’ to the power of ‘n’ using a while loop.

Ans: **Step 1:** Input integer number n and x.

**Step 2:** Initialize a function **double** pow(**double** x, **int** n) {

**Step 3: if** (n < 0) **return** pow(1.0 / x, -n)

**Step 4**: **if** (n == 0) **return** 1.0

**Step 5:** **if** (n == 1) **return** x

**Step 6:** **if** (n % 2 == 0) **return** pow(x \* x, n / 2)

**Step 7:** Default it will **return** x \* pow(x \* x, (n - 1) / 2)

1. To print the multiples of 3 between 1 to 20.

Ans**: Step 1:** Initialize a variable i and j. Make j static with a value equal to 3.

**Step 2:** Initialize a variable n=20 or define a limit variable.

**Step 3:** Design a for loop as for(i=1;i<=n/3;i++)

**Step 4:** Return 3\*i

**Step 5:** Print the values received by the function and close the program

**ASSIGNMENT 5**

Open the Python IDLE and execute the following commands. Observe the output.

1. 10 + 15

“It will display the sum of 10 and 15 that is 25 after the clicking of enter key.”

1. Print(“Hello World”)

“It will print hello world after we press enter key”

1. 45-34

“It will give us difference of 45 and 34 after we press enter key.”

1. 8\*2

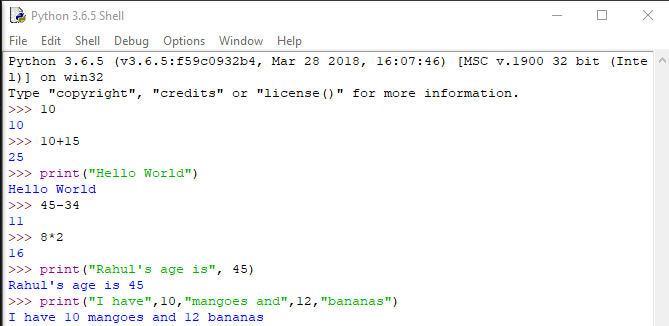
“It will display the products of 8 and 2 after we press enter key”

1. Print(“Rahul’s age is”,45)

It will give us the output as follows: “Rahul’s age is 45”

1. Print(“I have”,10,”mangoes and”,12,”bananas”)

I will give us the output as follows: “I have 10 mangoes and 12 bananas”



**ASSIGNMENT 6**

Open Python IDLE and execute the following commands. Observe the output.

1. emp\_number = 1233

Assign the value to the emp\_number variable.

1. print(“Emoployee Number:”, emp\_number)

It will print the output as follows:

Employee Number: 1233

1. emp\_salary = 16745.50

It will assign value to “emp\_salary” variable.

1. emp\_name = “Jerry Squaris”

It will assign string value to the “emp\_name” variable.

1. print(“Employee Salary and Name:”,emp\_salary,emp\_number)

It will print the output as follows:

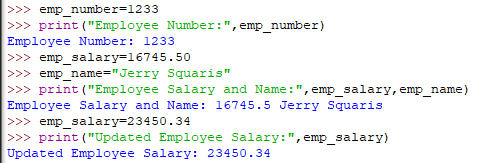
Employee Salary and Name: 16745.5 Jerry Squaris

1. emp\_salary = 23450.34

It will assign new value to same “emp\_salary” variable.

1. print(“Upadted Employee Salary:”,emp\_salary)

It will print the output as follows:  
“Updated Employee Salary: 23450.34”



**ASSIGNMENT 7**

Execute the following Python statements in IDLE and observe the output:

1. customer\_id=101

Assign the value to a variable named as customer\_id.

1. type(customer\_id)

Give the datatype of the variable taken as an argument.

1. customer\_name="John"

Assign the value to a variable named as customer\_name.

1. type(customer\_name)

Give the datatype of the variable taken as an argument.

1. bill\_amount=675.45

Assign the value to a variable named as bill\_amount.

1. type(bill\_amount)

Give the datatype of the variable taken as an argument.

1. x=5.3+0.9j

Assign the value to a variable named as x

1. type(x)

Give the datatype of the variable taken as an argument.

1. print(customer\_id,customer\_name,bill\_amount)

It will give output as follows:

1. print(x.real)

“It will print the real part of the complex number”

1. print(x.imag+3)

“It will increase the imaginary part by 3 and print it”

1. Flag=True

“It will store boolean variable in the Flag”

1. Type(Flag)

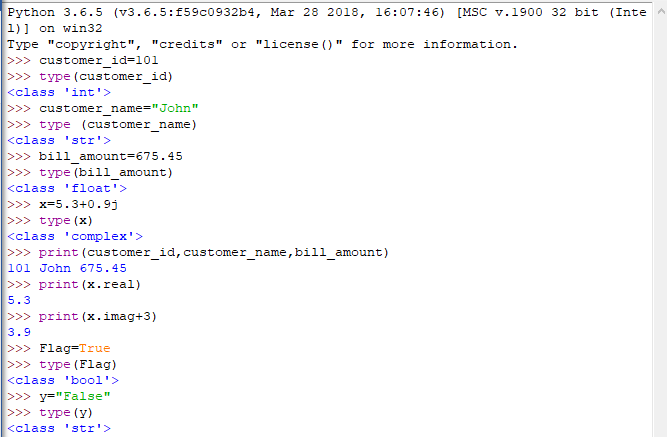
“It will give us the name of the datatype of the variable to which it belongs”

1. Y=”Flag”

“It will store the string variable in the variable named as y”.

1. Type(y)

“It will give us the name of the datatype to which it will belong”



**ASSIGNMENT 8**

In a retail application, shopkeeper wants to keep a track of following details of a customer. Sample values are

provided.

•bill\_id = 101

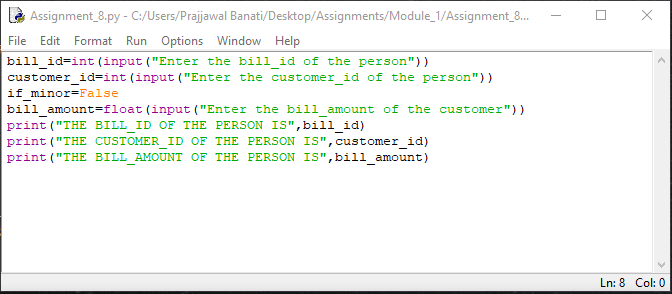
•customer\_id = 1001

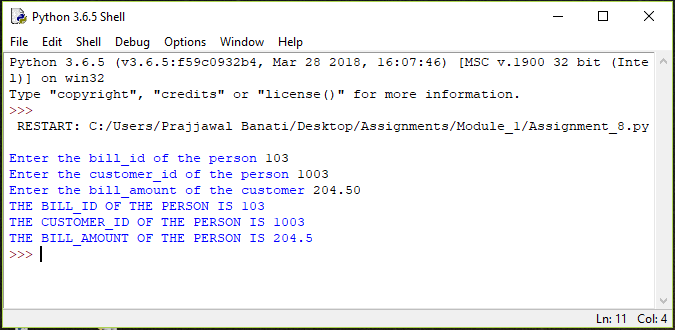
•customer\_name = "Rahul"

•if\_minor = False

•bill\_amount = 2000.50

Write a python program to store the details and display them





**ASSIGNMENT 9**

Execute the following commands and observe the usage of diﬀerent types of commenting styles.

i = 10

# creates an integer variable. This is a single line comment.

print("i =", i)

# prints 10

'''

Below code creates a Boolean variable in Python

(This is a multiple line comment)

'''

s = True

print("s =", s)

#prints True, Here, s is a Boolean variable with value True

"""

Below code assigns string data to variable 's'. Data type of variable can change during execution,

Hence, Python supports Dynamic Semantics.

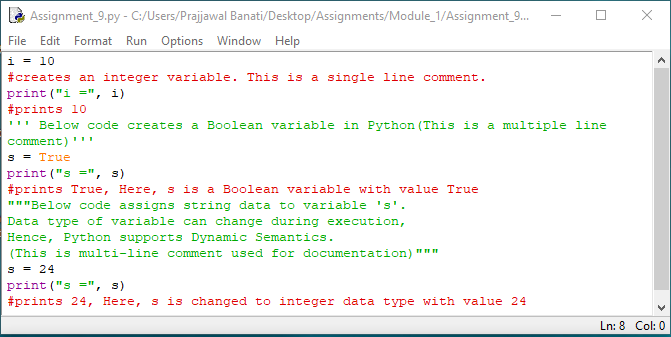
(This is multi-line comment used for documentation)

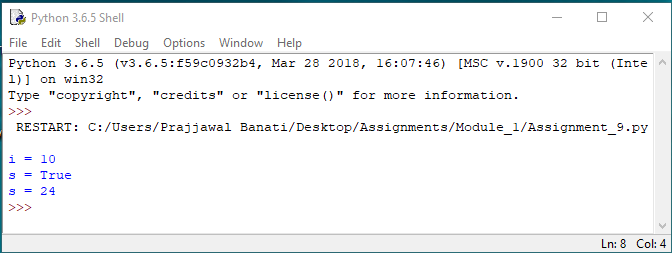
"""

s = 24

print("s =", s)

#prints 24, Here, s is changed to integer data type with value 24





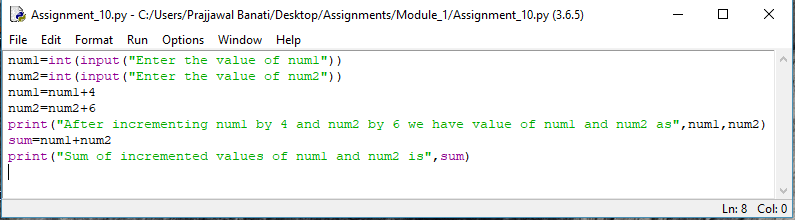
**ASSIGNEMENT 10**

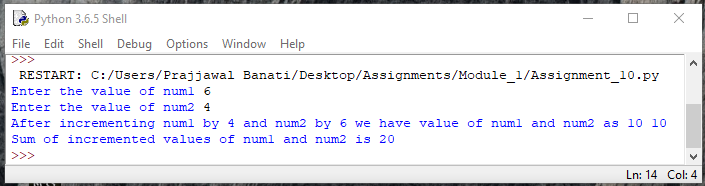
Write a Python program for the following requirements:

• Prompt the user to input two numbers num1 and num2

• Increment num1 by 4 and num2 by 6

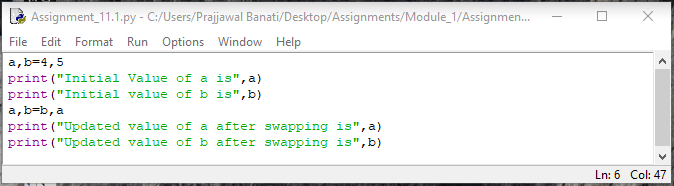
• Find and print the sum of new values of num1 and num2

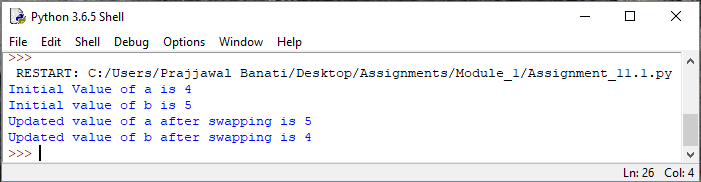




**ASSIGNMENT 11**

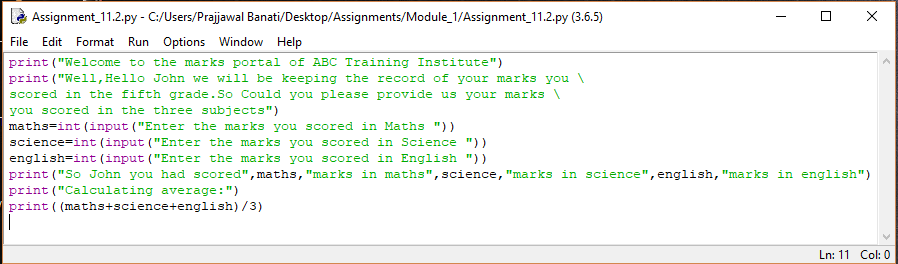
1. Consider two variables 'a' and 'b' in Python such that a = 4 and b = 5. Swap the values of 'a' and 'b' without using a temporary variable. Print the values of 'a' and 'b' before and after swapping

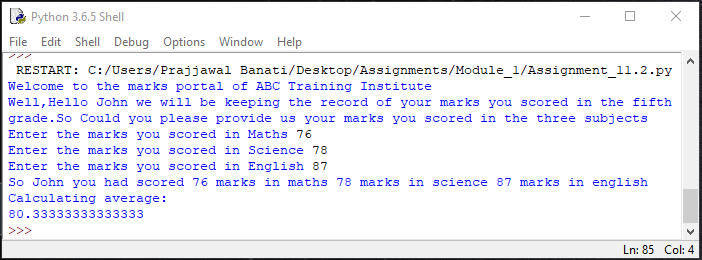




1. Consider the scenario of processing marks of a student in ABC Training Institute. John, the student of ﬁfth grade takes exams in three diﬀerent subjects. Create three variables to store the marks obtained by John in three

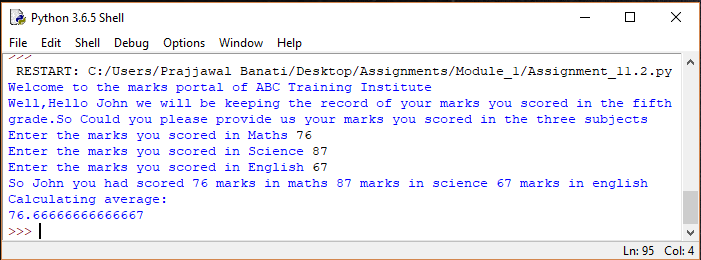
subjects. Find and display the average marks scored by John.



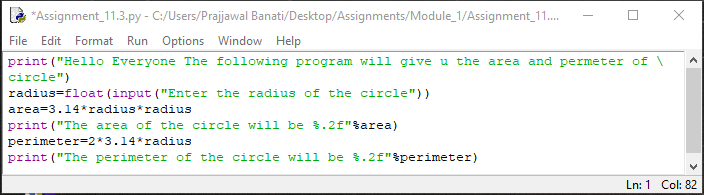


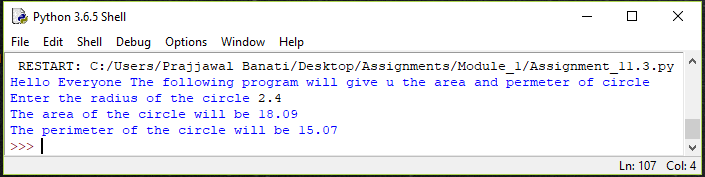
Now change the marks in one of the subjects and observe the output. Did the value of average change?

Ans: YES, AS NOW NEW VALUES WILL BE ASSIGNED TO SAME VARIABLES.



1. Given the value of radius of a circle, write a Python program to calculate the area and perimeter of the circle. Display both the values.





1. The ﬁnance department of a company wants to compute the monthly pay of its employees. Monthly pay should be calculated as mentioned in the formula below. Display all the employee details.

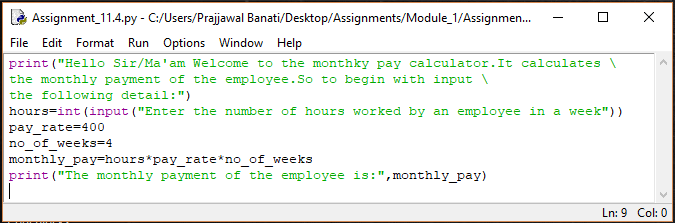
Monthly Pay = Number of hours worked in a week \* Pay rate per hour \* No. of weeks in a month

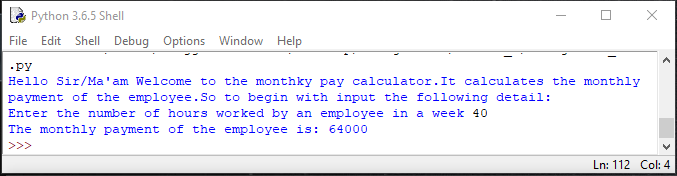
• The number of hours worked by the employee in a week should be considered as 40

• Pay rate per hour should be considered as Rs.400

• Number of weeks in a month should be considered as 4

Write a Python program to implement the above real world problem.





**ASSIGNEMENT 12**

Identify the sections of the given program where the coding standards are not followed and correct them.

1. itemNo= 1005

Ans: Firstly, all the letters should be in lowercase and secondly a combination of two words should be separated by \_ mark. And also there must be spaces between the binary operators. So the right representation of the variable is:

item\_no = 1005

1. unitprice = 250

Ans: Firstly we need underscore mark to separate the two words, So the right representation is

unit\_price = 250

1. quantity=2

Ans: Space between the binary operators

quantity = 2

1. amount=quantity\*unitprice

Ans: amount = quantity \* unit\_price

1. print(“Item No:”itemNo)

Ans: Firstly there must be space after the “Item No: ” so that it makes easy for the user to read the item no. and does not mix anything and one more thing itemNo should be all in lowercase and also spaced by underscore mark (‘\_’) , So the correct representation is

Print(“Item No: ”item\_no)

1. print(“Bill Amount:”,amount)

Ans: The correct representation is:

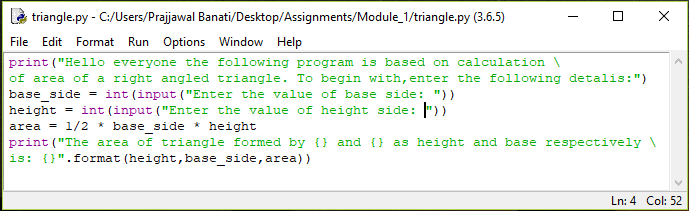
Print(“Bill Amount: ”,amount)

**ASSIGNMENT**

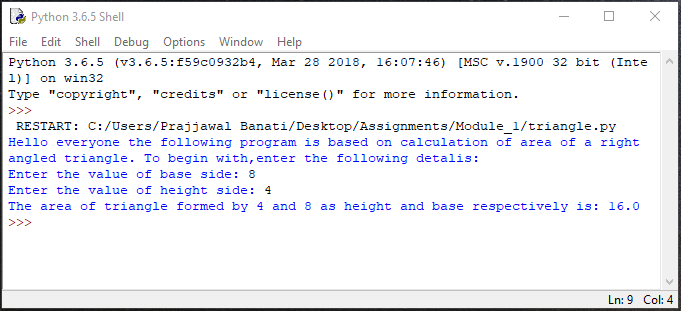
1. Create a file in PYTHON Idle named as triangle.py



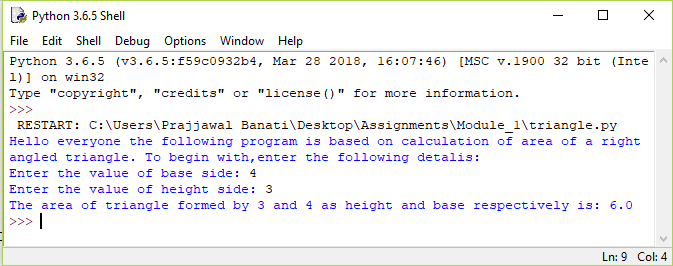
1. Write a Python program to calculate and print the area of the triangle. Prompt the user to input the values for base and height of the triangle.



1. Execute the program(use 'Run Module' under 'Run' tab) and observe the output.

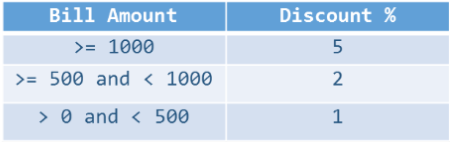


1. Close the ﬁle, open it again and execute it once more with diﬀerent values. Observe the output.

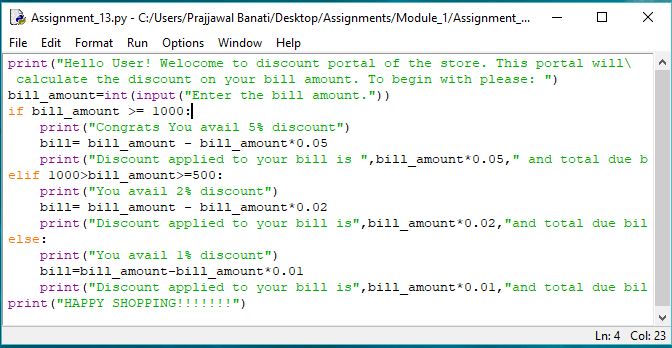


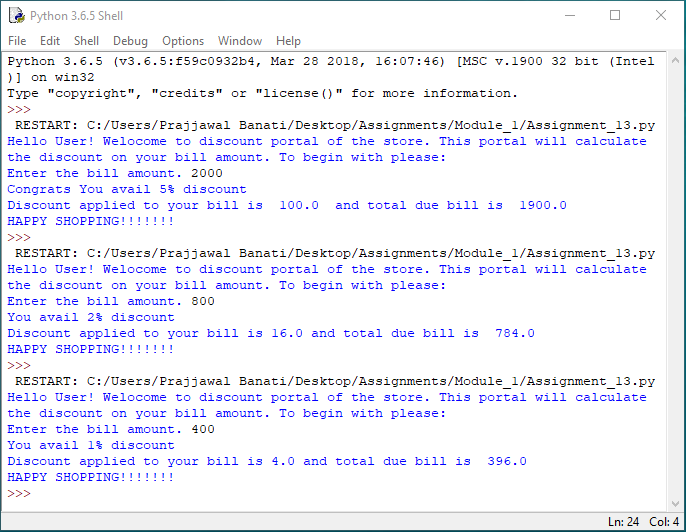
**ASSIGNMENT 13**

1. Consider the scenario of retail store management again. The store provides discount for all bill amounts based on the criteria below:

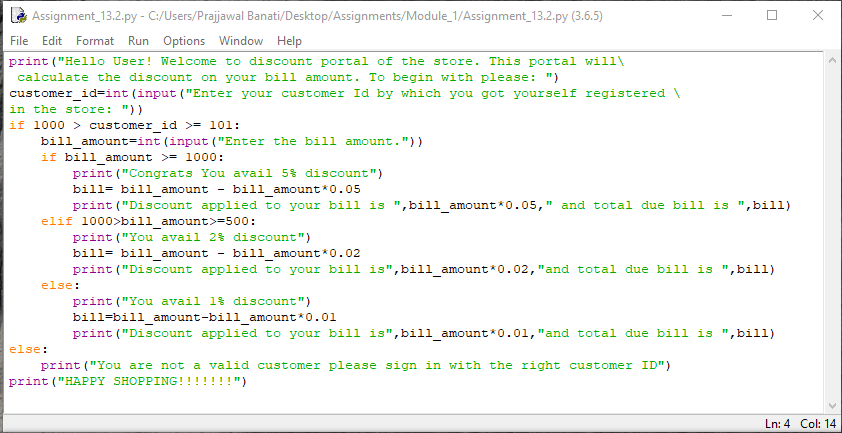


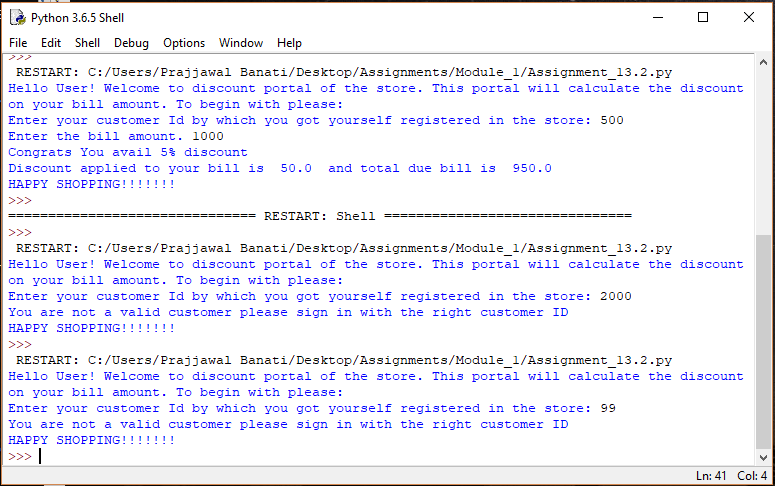
Write a Python program to ﬁnd the net bill amount after discount. Observe the output with diﬀerent values of bill amount. Assume that bill amount will be always greater than zero:





1. Extend the above program to validate the customer id. Customer ids in the range of 101 and 1000 (both inclusive) should only be considered valid.

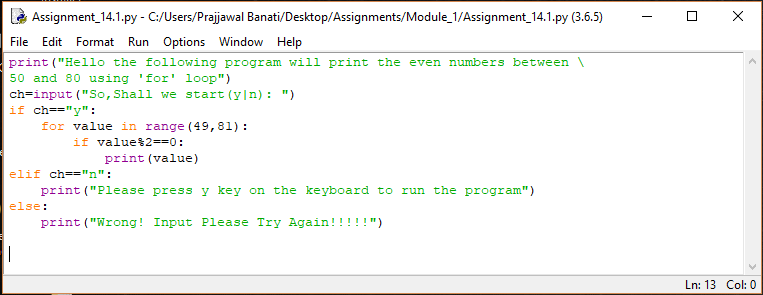


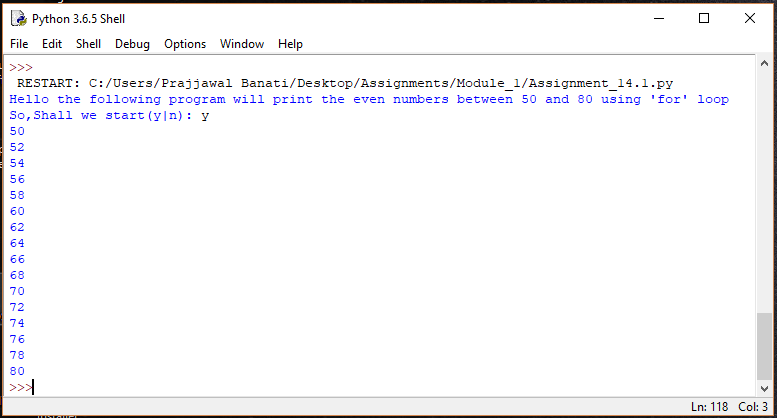


**ASSIGNMENT 14**

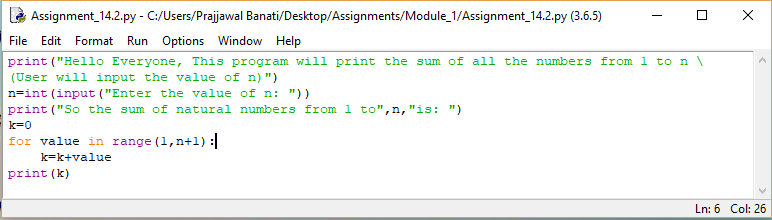
Implement the following in Python:

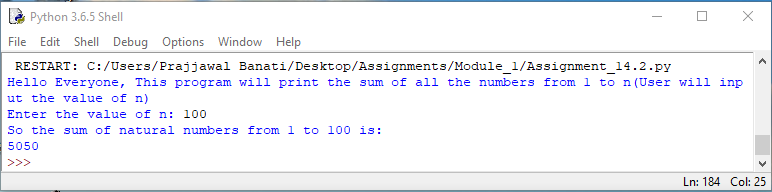
1. Display all even numbers between 50 and 80 (both inclusive) using "for" loop.



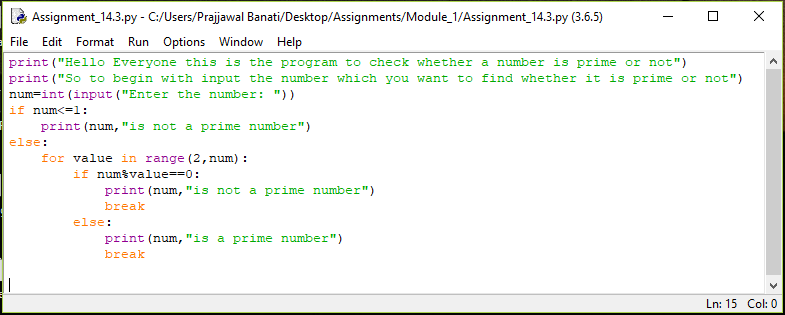


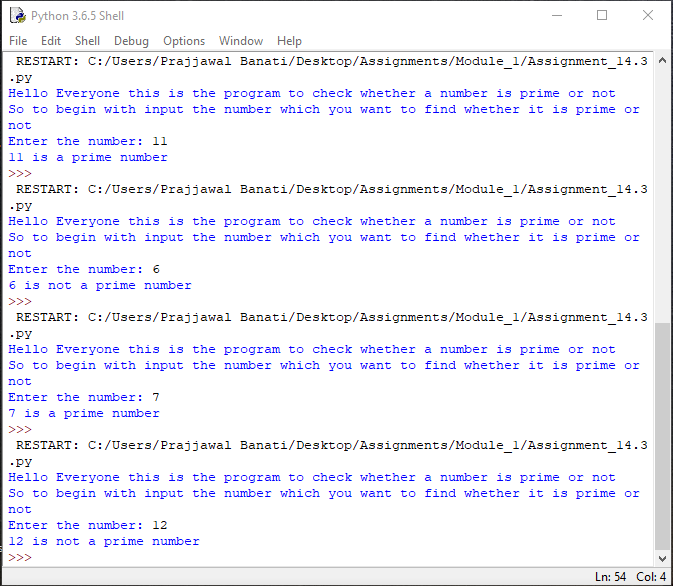
1. Add natural numbers up to n where n is taken as an input from user. Print the sum.



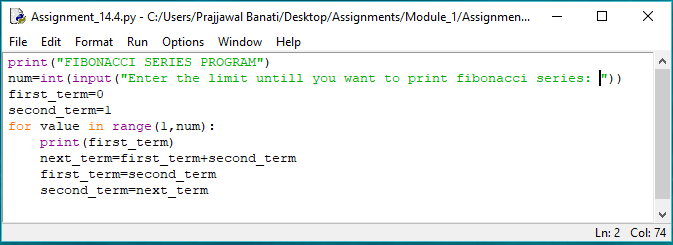


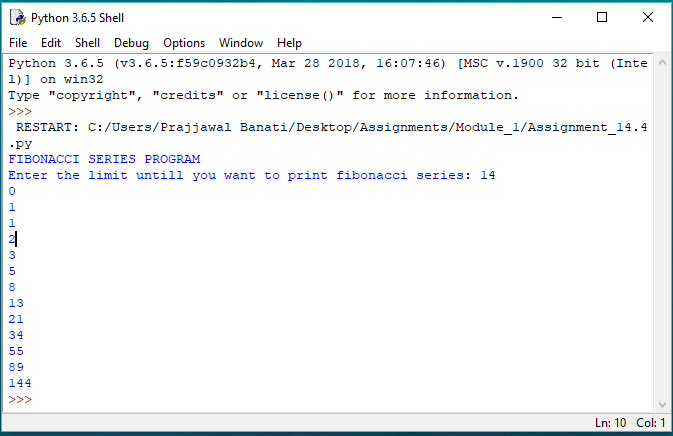
1. Prompt the user to enter a number. Print whether the number is prime or not.





1. Print Fibonacci series till nth term where n is taken as an input from user.





**ASSIGNMENT 15**

Create four string variables a, b, c, d

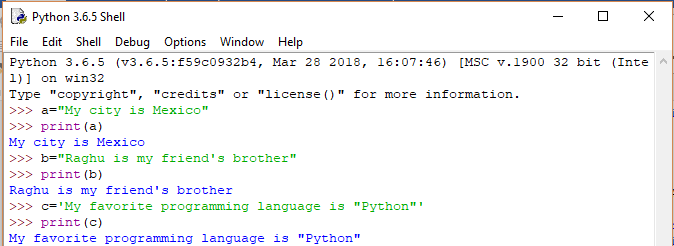
to store the following values and display them:

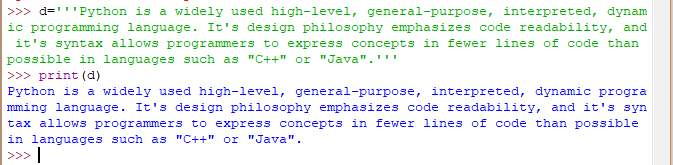
• My city is Mexico

• Raghu is my friend's brother

• My favorite programming language is "Python"

• Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. It's design philosophy emphasizes code readability, and it's syntax allows programmers to express concepts in fewer lines of code than possible in languages such as "C++" or "Java".





**ASSIGNMENT 16**

Accept a string as an input from the user. Check if the accepted string is palindrome or not.

• If the string is palindrome, print "String is palindrome", otherwise print "String is not palindrome".

• Also print the actual and the reversed strings.

Note – Ignore the case of characters.

Hint – A palindrome string remains the same if the characters of the string are reversed.

Ans:

Code:

print("Hello Everyone! This is the program of checking whether \

the strings are palindrome or not")

string=input("Input the string: ")

print("The original string is:",string)

#using an empty string to concatenate it while reversing

reverse\_string=""

a=len(string)

#Reversing the string using a for loop

for i in range(-1,-a-1,-1):

#See here "b" will start storing the reverse string

reverse\_string=reverse\_string+string[i]

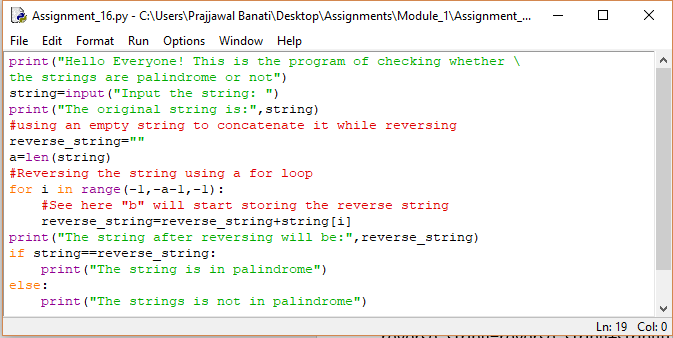
print("The string after reversing will be:",reverse\_string)

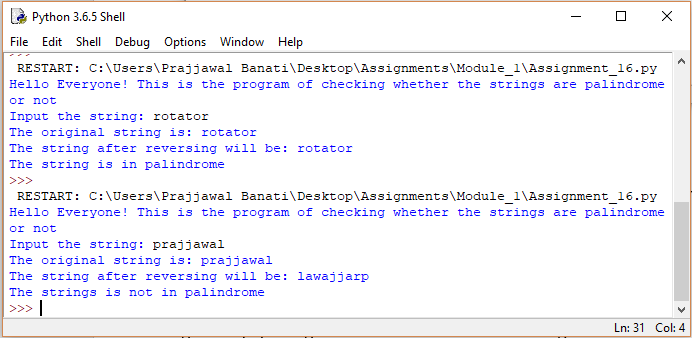
if string==reverse\_string:

print("The string is in palindrome")

else:

print("The strings is not in palindrome")





**ASSIGNMENT 17**

Accept two strings 'string1' and 'string2' as an input from the user. Generate a resultant string, such that it is a concatenated string of all upper case alphabets from both the strings in the order they appear. Print the actual and the resultant strings.

Note: Each character should be checked if it is a upper case alphabet and then it should be concatenated to the

resultant string.

Sample Input: string1: I Like C

string2: Mary Likes Python

Output: ILCMLP

Code:

print("Hello Everyone! The following program will take two string as an input \

from your and will concatenate all the capital letters from both the strings")

#Storing a string of capital letters in compiler to compare it with strings 1 and 2

default="ABCDEFGHIJKLMNOPQRSTUVWXYZ"

#Storing two Empty strings in which we insert the capital letter string

capital\_string\_1=""

capital\_string\_2=""

#Store length

a=len(default)

string\_1 = input("Enter string 1: ")

b=len(string\_1)

#Comparing each letter of string with the alphabet string

for i in range(0,b,1):

for j in range(0,a,1):

if string\_1[i]==default[j]:

capital\_string\_1 = capital\_string\_1 + string\_1[i]

print("The capital string in string\_1 is",capital\_string\_1)

string\_2 = input("Enter string 2: ")

b=len(string\_2)

for i in range(0,b,1):

for j in range (0,a,1):

'''If the element gets equal to any alphabet in default string it will get

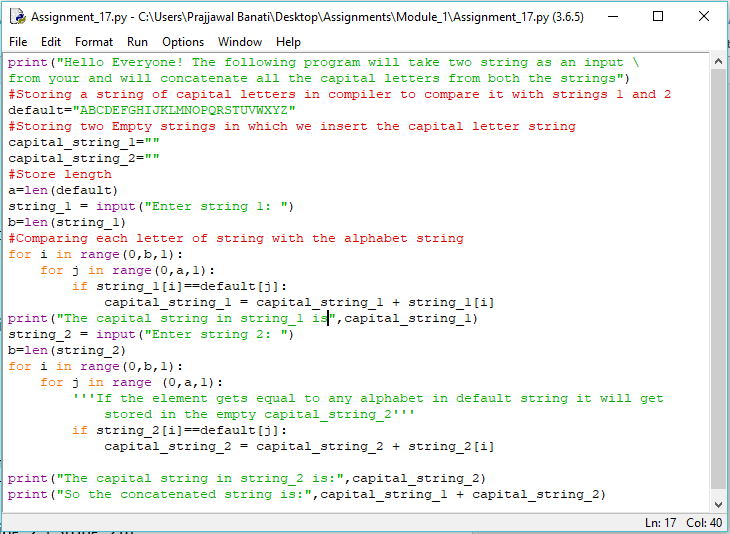
stored in the empty capital\_string\_2'''

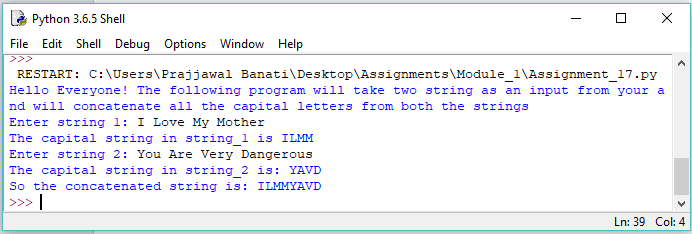
if string\_2[i]==default[j]:

capital\_string\_2 = capital\_string\_2 + string\_2[i]

print("The capital string in string\_2 is:",capital\_string\_2)

print("So the concatenated string is:",capital\_string\_1 + capital\_string\_2)





**ASSIGNMENT 19**

1. Write a Python program to accept a string 'accepted\_string'. Generate a resultant string 'resultant\_string' such that 'resultant\_string' should contain all characters at the even position of 'accepted\_string'(ignoring blank spaces). Display “resultant\_string' in reverse order.

accepted\_string: An apple a day keeps the doctor away

resultant\_string: Aapedyepteotrwy

expected\_output: ywrtoetpeydepaA

Code:

print("Hello Everyone the following program will take a string from you \

then it would select only the even places of the string ignoring the spaces \

and will give you a string in the reversed order")

resultant\_string=""

expected\_output=""

accepted\_string = input("Write a string: ")

print(accepted\_string)

#replace funtion is used to replace any variable or a space in the string

#So here we replaced the spaces to get the correct position of string

accepted\_string=accepted\_string.replace(" ","")

a=len(accepted\_string)

for i in range(0,a,1):

if (i%2==0):

resultant\_string=resultant\_string+accepted\_string[i]

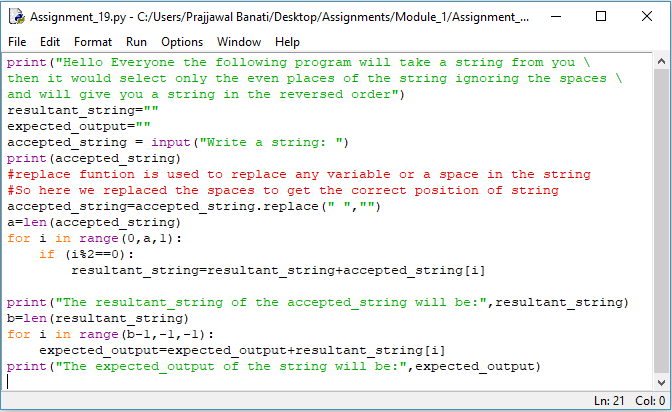
print("The resultant\_string of the accepted\_string will be:",resultant\_string)

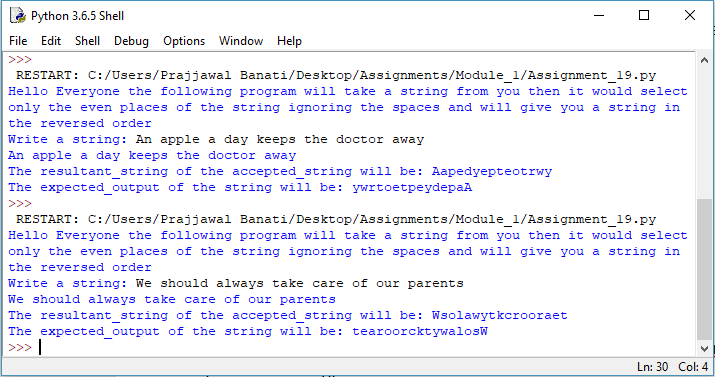
b=len(resultant\_string)

for i in range(b-1,-1,-1):

expected\_output=expected\_output+resultant\_string[i]

print("The expected\_output of the string will be:",expected\_output)





**ASSIGNMENT 20**

1. Write a Python program to generate ﬁrst 'n' Fibonacci numbers where 'n' is accepted as an input from the user. Store the generated Fibonacci numbers in a list and display the output.

Sample input: 5

Sample output: [0, 1, 1, 2, 3]

Ans: print("Fibonacci numbers in a list:")

print("Print the fibonacci series in a list.")

num=int(input("Enter the limit untill you want to print fibonacci series: "))

fibonacci\_list =[]

first\_term=0

second\_term=1

for i in range(0,num):

fibonacci\_list.insert(i,first\_term)

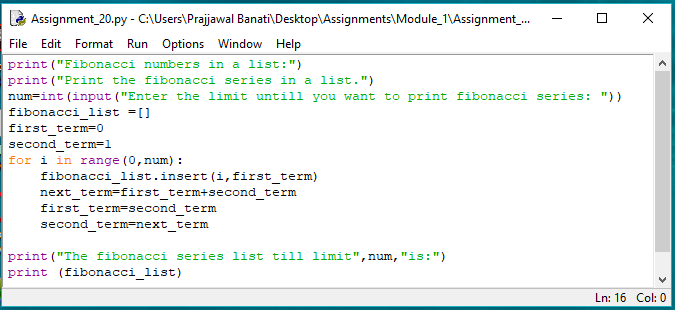
next\_term=first\_term+second\_term

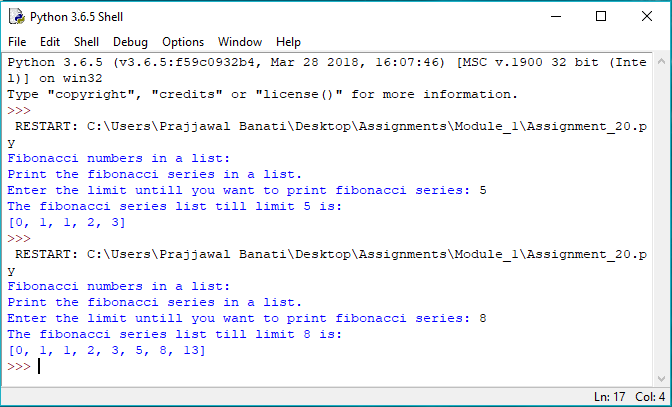
first\_term=second\_term

second\_term=next\_term

print("The fibonacci series list till limit",num,"is:")

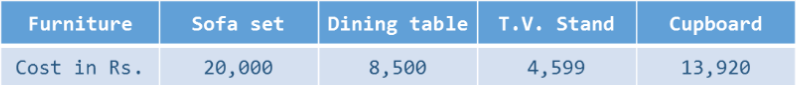
print (fibonacci\_list)





**ASSIGNMENT 21**

1. The "Variety Retail Store" sells diﬀerent varieties of Furniture to the customers. The list of furniture available with its respective cost is given below:



The furniture and its corresponding cost should be stored as a list. A customer can order any furniture in any quantity (the name and quantity of the furniture will be provided). If the required furniture is available in the furniture list (given above) and quantity to be purchased is greater than zero, then bill amount should be calculated. In case of invalid values for furniture required by the customer and quantity to be purchased, display appropriate error message and consider bill amount to be 0. Initialize required furniture and quantity with diﬀerent values and test the results. Write a Python program to calculate and display the bill amount to be paid by the customer based on the furniture bought and quantity purchased.

Ans:

print('Hello Everyone! Welcome to "Variety retail store". Here is the \

list of the items and there prices.')

furniture=["sofa set", "dining table", "T.V. stand", "cupboard"]

cost=[20000, 8500, 4599, 13920]

f=input("Enter the furniture you want to buy:")

q=int(input("Enter quantity:"))

c=-1

flag=0

for x in furniture:

c = c + 1

if(f==x):

flag = flag + 1

if(q>0):

bill = cost[c] \* q

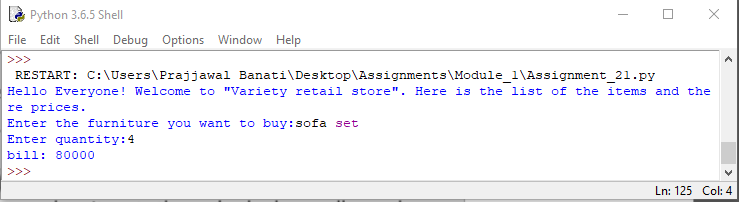
print("bill:",bill)

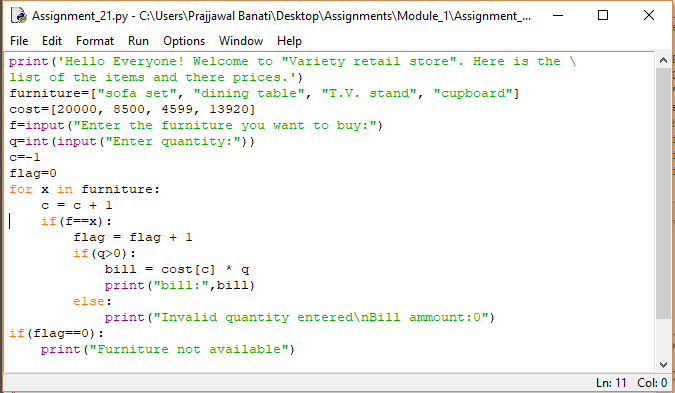
else:

print("Invalid quantity entered\nBill ammount:0")

if(flag==0):

print("Furniture not available")





**ASSIGNMENT 22**

Consider the list of courses opted by a Student "John" and available electives at ABC Training Institute:

courses = ("Python Programming", "RDBMS", "Web Technology", "Software Engg.")

electives = ("Business Intelligence", "Big Data Analytics")

Write a Python Program to satisfy business requirements mentioned below:

1. List the number of courses opted by John.

2. List all the courses opted by John.

3. John is also interested in elective courses mentioned above. Print the updated tuple including electives.

Ans:

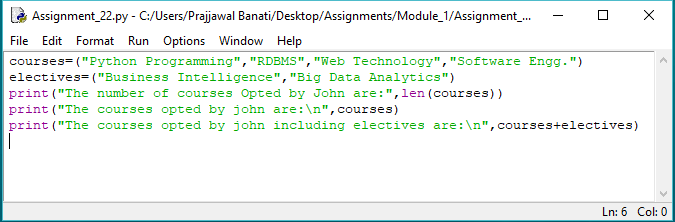
courses=("Python Programming","RDBMS","Web Technology","Software Engg.")

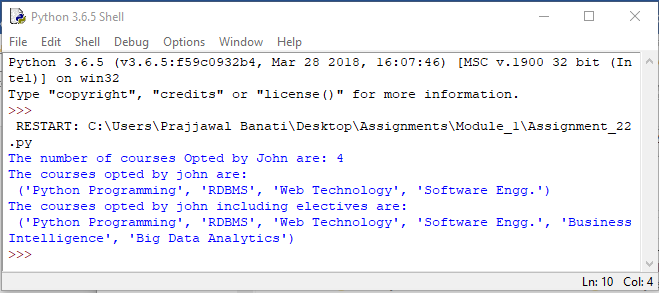
electives=("Business Intelligence","Big Data Analytics")

print("The number of courses Opted by John are:",len(courses))

print("The courses opted by john are:\n",courses)

print("The courses opted by john including electives are:\n",courses+electives)





**ASSIGNMENT 23**

Given below is a dictionary 'customer\_details' representing customer details from a Retail Application. Customer Id

is the key and Customer Name is the value.

customer\_details = { 1001 : "John", 1004 : "Jill", 1005: "Joe", 1003 : "Jack" }

Write Python code to perform the operations mentioned below:

a) Print details of customers.

b) Print number of customers.

c) Print customer names in ascending order.

d) Delete the details of customer with customer id = 1005 and print updated dictionary.

e) Update the name of customer with customer id = 1003 to "Mary" and print updated dictionary.

f) Check whether details of customer with customer id = 1002 exists in the dictionary.

Ans:

customer\_details={1001:"John",1004:"Jill",1005:"Joe",1003:"Jack"}

print("The details of the customer are:\n",customer\_details.values())

print("The number or the customer\_id of the customers is:\n",customer\_details.keys())

print("The details of the customers in ascending order:\n",sorted(customer\_details.values()))

del customer\_details[1005]

print("The details of the customer after deleting id=1005",customer\_details)

customer\_details[1003]="Mary"

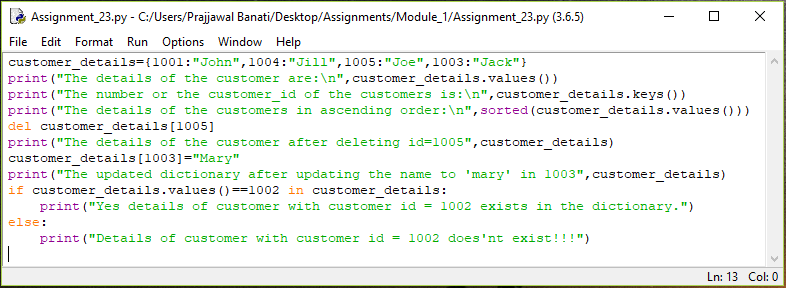
print("The updated dictionary after updating the name to 'mary' in 1003",customer\_details)

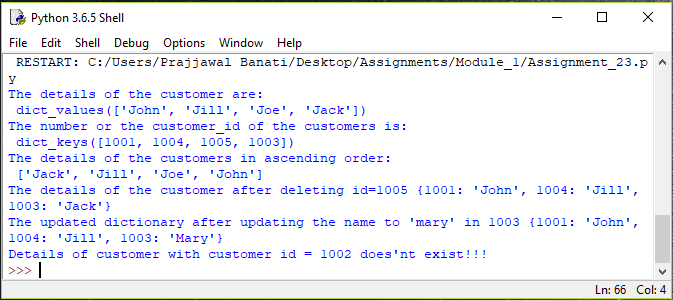
if customer\_details.values()==1002 in customer\_details:

print("Yes details of customer with customer id = 1002 exists in the dictionary.")

else:

print("Details of customer with customer id = 1002 does'nt exist!!!")





**ASSIGNMENT 24**

Consider a scenario from ABC Training Institute. The given table shows the marks scored by students of grade XI in

Python Programming course.



Write a Python program to meet the requirements mentioned below:

a. Display the name and marks for every student.

b. Display the top two scorers for the course.

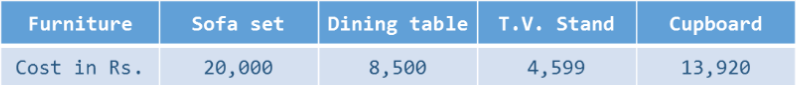
c. Display class average of this course.

Hint- Implement the solution using a dictionary.

Ans:

**ASSIGNMENT 25**

Consider the scenario from "Variety Retail Store" discussed in 'List' section. The list of furniture available with its respective cost is given below:



A customer can order any furniture in any quantity. If the required furniture is available in the furniture list(given above) and quantity to be purchased is greater than zero, then bill amount should be calculated. In case of invalid values for furniture required by the customer and quantity to be purchased, display appropriate error message and

consider bill amount to be 0. Initialize required furniture and quantity with diﬀerent values and test the results. Calculate and display the bill amount to be paid by the customer based on the furniture bought and quantity purchased. Implement the given scenario using:

1) List of tuples

2) Dictionary

Ans:

#tuple\_representation

furniture=("sofa set", "dining table", "T.V. stand", "cupboard")

cost=(20000, 8500, 4599, 13920)

f=input("Enter the furniture you want to buy:")

q=int(input("Enter quantity:"))

c=-1

flag=0

for x in furniture:

c = c + 1

if(f==x):

flag = flag + 1

if(q>0):

bill = cost[c] \* q

print("Furniture:",x,"\nquantity:",q)

print("bill:",bill)

else:

print("Invalid quantity entered\nBill ammount:0")

if(flag==0):

print("Furniture not available")

#dictionary\_representation

furniture={"sofa set":20000, "dining table":8500, "T.V. stand":4599, "cupboard":13920}

f=input("Enter the furniture you want to buy:")

q=int(input("Enter quantity:"))

#c=-1

flag=0

for key in furniture:

#c = c + 1

if(f==key):

flag = flag + 1

if(q>0):

bill = furniture[key] \* q

print("Furniture:",key,"\nquantity:",q)

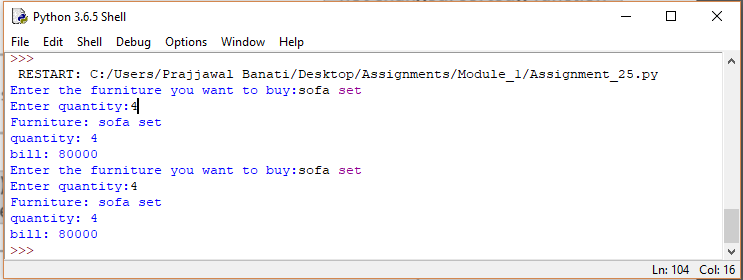
print("bill:",bill)

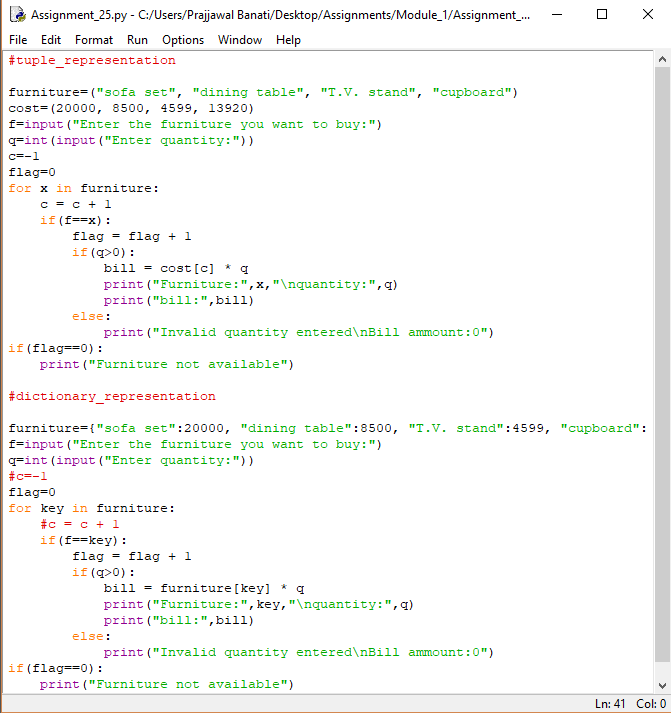
else:

print("Invalid quantity entered\nBill ammount:0")

if(flag==0):

print("Furniture not available")





**ASSIGNMENT 26**

Consider a scenario from ABC Training Institute. Given below are two Sets representing the names of students

enrolled for a particular course:

java\_course = {"John", "Jack", "Jill", "Joe"}

python\_course = {"Jake", "John", "Eric", "Jill"}

Write a Python program to list the number of students enrolled for:

1. Python course

2) Java course only

3) Python course only

4) Both Java and Python courses

5) Either Java or Python courses but not both

6) Either Java or Python courses

Ans:

java\_course={"John","Jack","Jill","Joe"}

python\_course={"Jake","John","Eric","Jill"}

print("The students who are a part of Python course are:",python\_course)

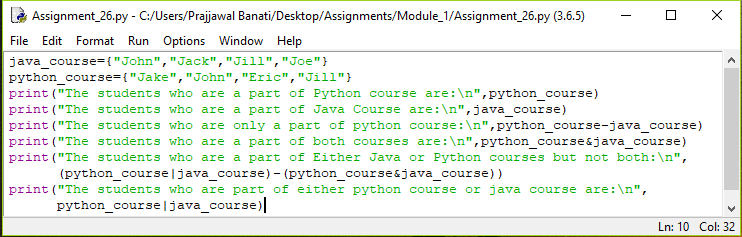
print("The students who are a part of Java Course are:",java\_course)

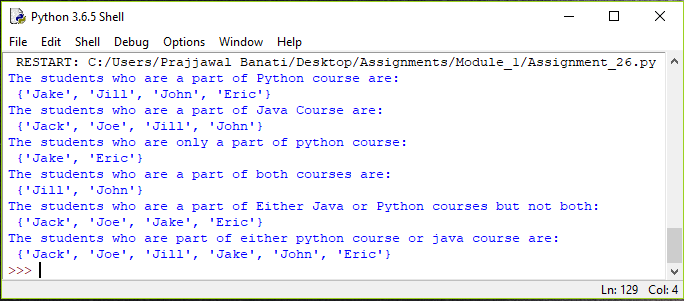
print("The students who are only a part of python course:",python\_course-java\_course)

print("The students who are a part of both courses are:",python\_course&java\_course)

print("The students who are a part of Either Java or Python courses but not both:",(python\_course|java\_course)-(python\_course&java\_course))

print("The students who are part of either python course or java course are:",python\_course|java\_course)





**ASSIGNMENT 27**

Using functions, re-write and execute Python program to:

1. Add natural numbers upto n where n is taken as an input from user.

Ans:

print("Hello Everyone, This program will print the sum of all the numbers from 1 to n \

(User will input the value of n)")

n=int(input("Enter the value of n: "))

def sum(n):

print("So the sum of natural numbers from 1 to",n,"is: ")

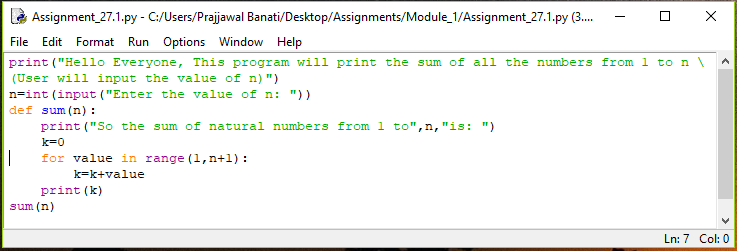
k=0

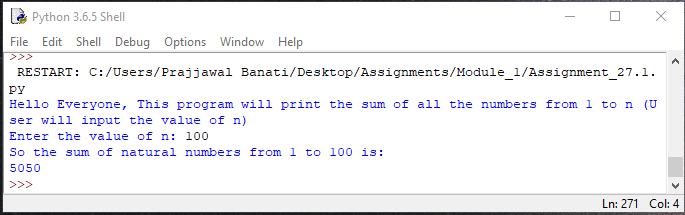
for value in range(1,n+1):

k=k+value

print(k)

sum(n)





1. Print Fibonacci series till nth term (Take input from user).

Ans:

print("FIBONACCI SERIES PROGRAM")

num=int(input("Enter the limit untill you want to print fibonacci series: "))

def fibonacci(num):

first\_term=0

second\_term=1

for value in range(1,num):

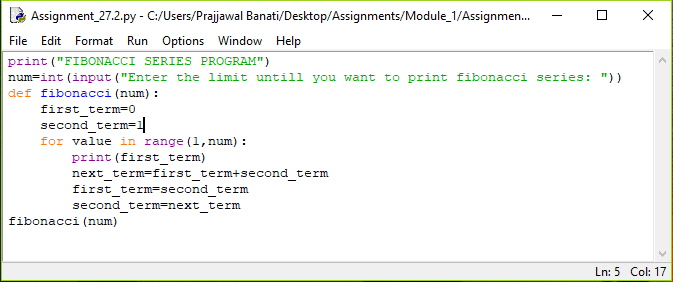
print(first\_term)

next\_term=first\_term+second\_term

first\_term=second\_term

second\_term=next\_term

fibonacci(num)





**ASSIGNMENT 28**

At an airport, a traveler is allowed entry into the ﬂight only if he clears the following checks:

1.Baggage Check

2.Immigration Check

3.Security Check

The logic for the check methods are given below:

check\_baggage (baggage\_weight)

•returns True if

baggage\_weight is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.

check\_immigration (expiry\_year)

•.returns True if expiry\_year is greater than or equal to 2001 and less than or equal to 2025. Otherwise returns False.

check\_security(noc\_status)

•.returns True if noc\_status is 'valid' or 'VALID', for all other values return False.

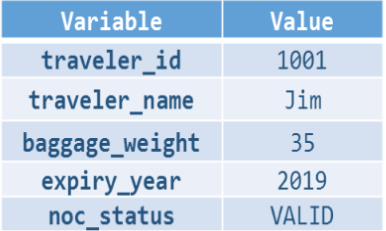
traveler()

•.Initialize the traveler Id and traveler name and invoke the functions check\_baggage(), check\_immigration() and check\_security() by passing required arguments.

•.Refer the table below for values of arguments.

•.If all values of check\_baggage(), check\_immigration() and check\_security() are true,

display traveler\_id and traveler\_name display "Allow Traveler to ﬂy!"



Otherwise,display traveler\_id and traveler\_name display "Detain Traveler for Re-checking!“ Invoke the traveler() function. Modify the values of diﬀerent variables in traveler() function and observe the output.

Ans:

print("Hello and welcome to the airport portal.To continue please enter the\

traveller\_id and traveler\_name")

def check\_baggage(baggage\_weight):

if 40>baggage\_weight>=0:

return True

else:

return False

def check\_immigration(expiry\_year):

if 2025>expiry\_year>=2001:

return True

else:

return False

def check\_security(noc\_status):

if noc\_status=="valid" or "VALID":

return True

else:

return False

def traveler():

traveler\_id=int(input("Enter the traveler\_id: "))

traveler\_name=input("Enter the traveler name: ")

baggage\_weight=int(input("Enter the baggage weight of the traveller: "))

a=check\_baggage(baggage\_weight)

expiry\_year=int(input("Enter the expiry year of the passport: "))

b=check\_immigration(expiry\_year)

noc\_status=input("Enter the status of noc: ")

c=check\_security(noc\_status)

if a==True and b==True and c==True:

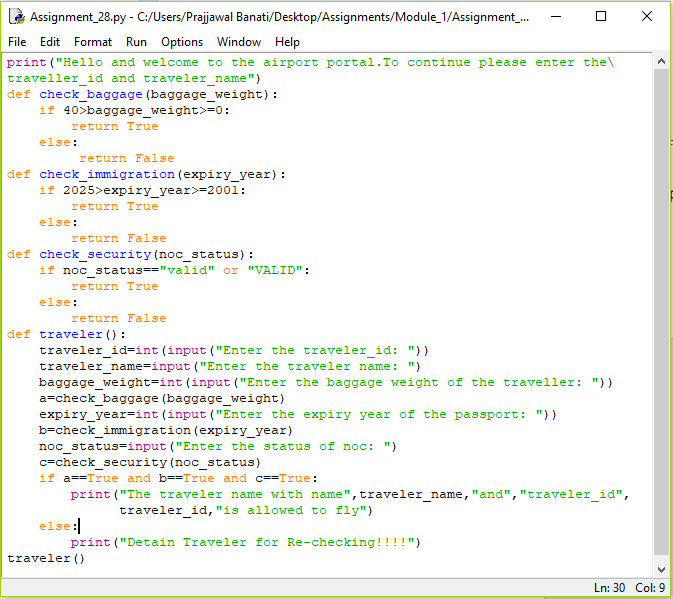
print("The traveler name with name",traveler\_name,"and","traveler\_id",

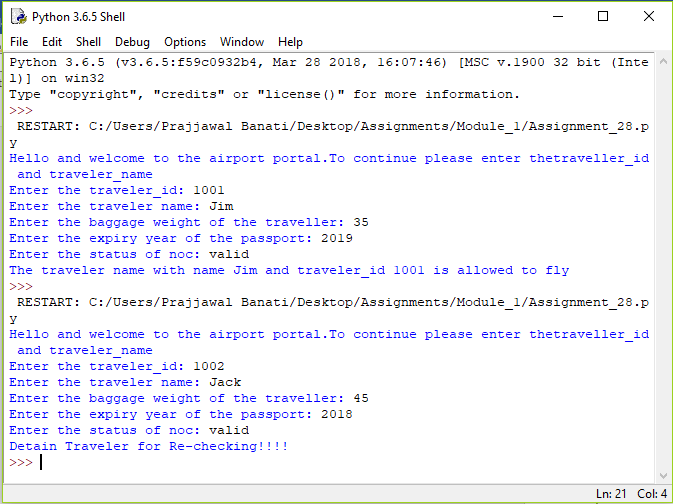
traveler\_id,"is allowed to fly")

else:

print("Detain Traveler for Re-checking!!!!")

traveler()





**ASSIGNMENT 29**

Consider the pseudo code for generating Fibonacci series using Recursion:

FIBO (number)

1. if (number = 0) then

2. return (0)

3. else if (number = 1) then

4. return (1)

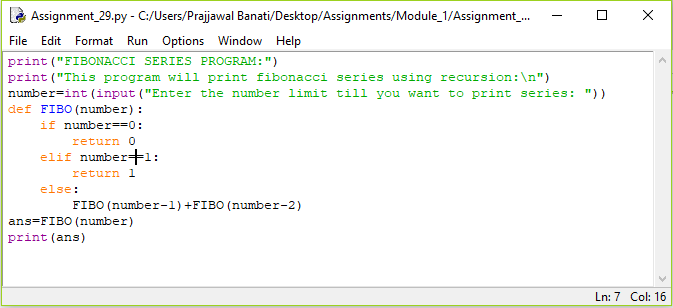
5. else

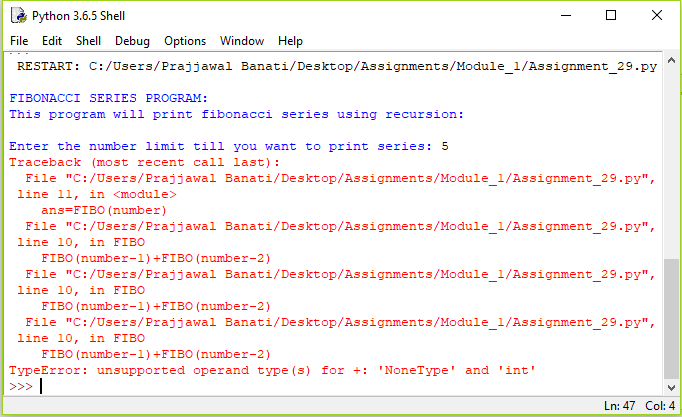
6. return FIBO(number - 1) + FIBO(number - 2)

7. end if

Write a program in Python to implement the same using Recursion and execute it in Eclipse. Print appropriate error message if the user enters negative number as input.

Ans:





**ASSIGNMENT 30**

Write a Python program to implement the following (Use Recursion):

1. Print ﬁrst 'n' multiples of 3, where 'n' is taken as an input from the user. The multiples should be printed from ﬁrst to last

Ans:

print("The following program will print the multiples of 3 using recursion")

n=int(input("Enter the value of n: "))

multiple=[]

def multiples(n):

if n==1:

return multiple.append(3)

else:

multiple.append(n\*3)

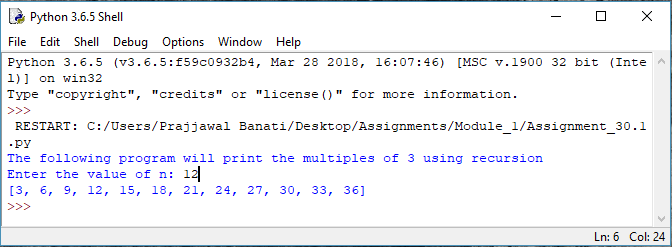
n-=1

return multiples(n)

multiples(n)

print(multiple[::-1])





1. Reverse a string. Print the original and reversed string.

Ans:

print("Hello Everyone! This is the program of printing the reverse string")

string=input("Input the string: ")

print("The original string is:",string)

a=len(string)

concatenated=""

b=a-1

def reverse\_string(b,string):

global concatenated

if b<0:

return

else:

concatenated=concatenated+string[b]

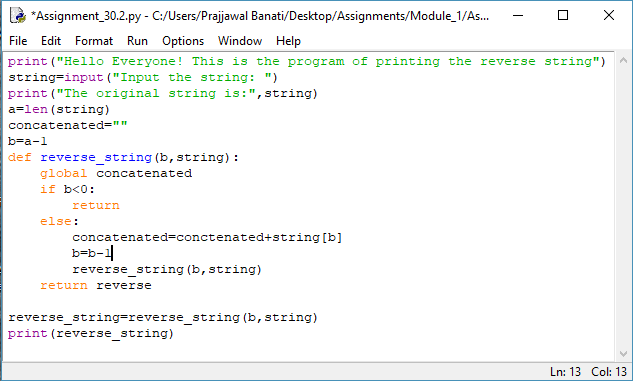
b=b-1

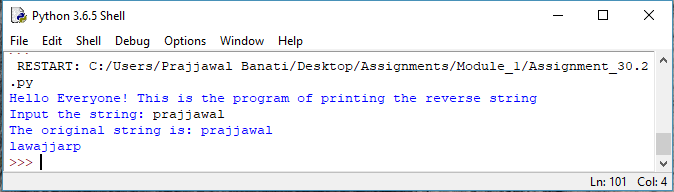
reverse\_string(b,string)

return conatenated

reverse\_string=reverse\_string(b,string)

print(reverse\_string)





1. Check if the given string is palindrome. If yes, print "String is palindrome" otherwise print "String is not palindrome".

Ans:

print("Hello Everyone! This is the program of checking whether the string is \

palindrome or not")

string=input("Input the string: ")

print("The original string is:",string)

a=len(string)

concatenated=""

b=a-1

def reverse\_string(b,string):

global concatenated

if b<0:

return

else:

concatenated=concatenated+string[b]

b=b-1

reverse\_string(b,string)

return concatenated

reverse\_string=reverse\_string(b,string)

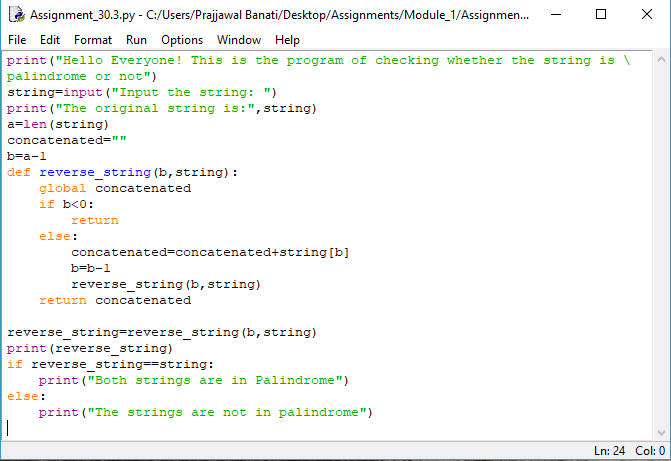
print(reverse\_string)

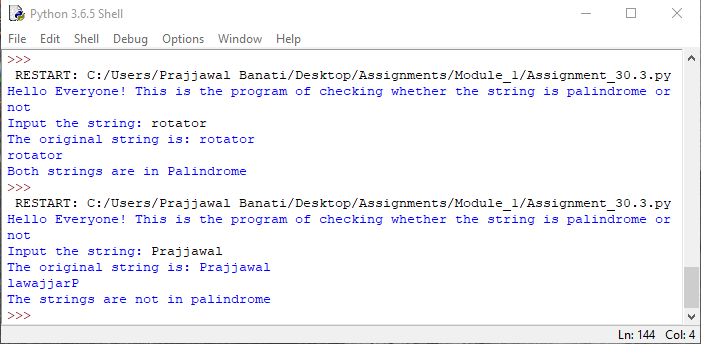
if reverse\_string==string:

print("Both strings are in Palindrome")

else:

print("The strings are not in palindrome")





**ASSIGNMENT 31**

1. Write a Python program to:

1. read a ﬁle.

2. add backslash (\) before every double quote in the ﬁle contents.

3. write it to another ﬁle in the same folder.

4. print the contents of both the ﬁles.

For example:

If the ﬁrst ﬁle is 'TestFile1.txt' with text as:

Jack said, "Hello Pune".

The output of the ﬁle 'TestFile2.txt' should be:

Jack said,\"Hello Pune\".

Ans:

1. **Making of the file testfile1.txt**

TestFile1=open("C:\\Users\\Prajjawal Banati\\Desktop\\Assignments\\Module\_1\\TestFile1.txt","w")

TestFile1.write(“Hello Pune”)

TestFile.close()

1. **Copying to another file named as testfile2.txt**

TestFile1=open("C:\\Users\\Prajjawal Banati\\Desktop\\Assignments\\Module\_1\\TestFile1.txt","r+")

TestFile2=open("C:\\Users\\Prajjawal Banati\\Desktop\\Assignments\\Module\_1\\TestFile2.txt","w")

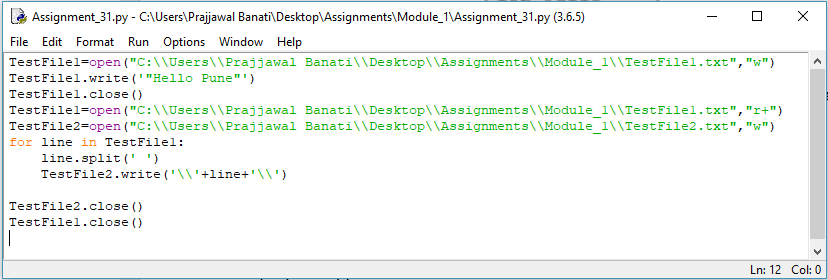
for line in TestFile1:

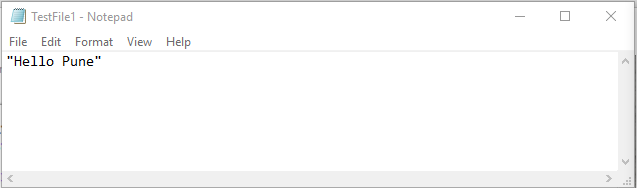
line.split(' ')

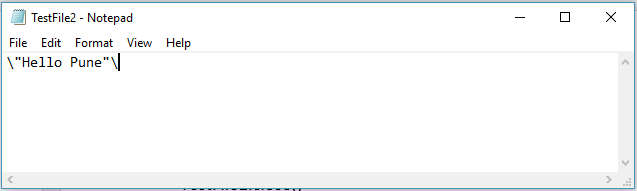
TestFile2.write('\\ '+line+' \\')

TestFile2.close()

TestFile1.close()

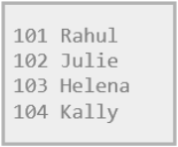






**ASSIGNMENT 32**

Consider a ﬁle 'courses.txt' in D Drive with the following details:



Write a program to read the ﬁle and store the courses in Python variables as a:

1. Dictionary ( Sample - {0: 'Java', 1: 'Python', 2:'Javascript' 3: 'PHP'} )
2. List ( Sample - ['Java', 'Python', 'Javascript', 'PHP'] )

Ans:

file=open("D:\\courses\\courses.txt","r")

List=[]

for line in file:

tokens=line.split()

List=List+tokens

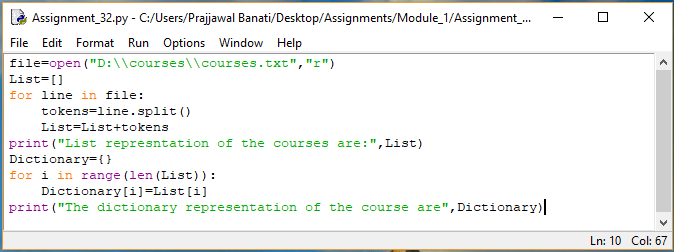
print("List represntation of the courses are:",List)

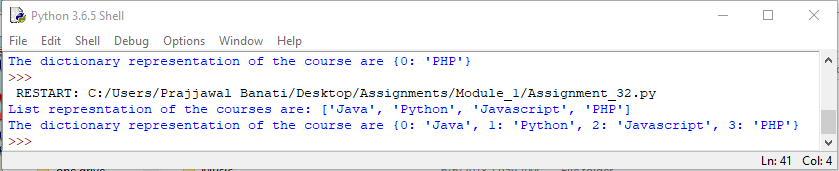
Dictionary={}

for i in range(len(List)):

Dictionary[i]=List[i]

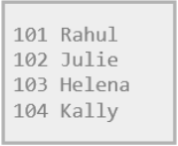
print("The dictionary representation of the course are",Dictionary)





**ASSIGNMENT 33**

Consider a ﬁle 'student\_details.txt' in D Drive with the details of students in ABC institute – student id and name:



Write a program to read the ﬁle and store the student records in Python variable as:

1) List of lists

2) List of dictionaries

Ans:

file=open("D:\\courses\\Student\_Details.txt","r")

List=[]

Dictionary={}

for line in file:

tokens=line.split()

List=List+tokens

print("The list represenation of the rollno and mame is",List)

for i in range(1,len(List),2):

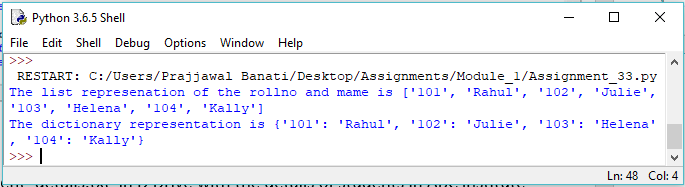
key=List[i-1]

Dictionary[key]=List[i]

i=i+1

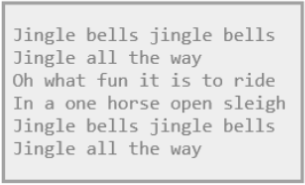
print("The dictionary representation is",Dictionary)





**ASSIGNMENT 34**

Consider a ﬁle 'rhyme.txt' in D Drive with following text:



Write a Python program to count the words in the ﬁle using a dictionary (use space as a delimiter). Find unique words and the count of their occurrences(ignoring case). Write the output in another ﬁle "words.txt" at the same location

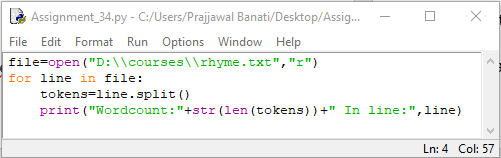
Ans:

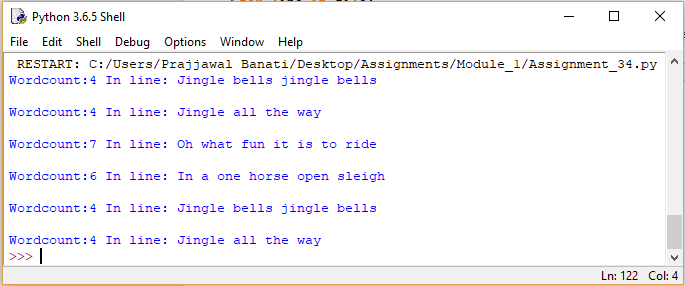
file=open("D:\\courses\\rhyme.txt","r")

for line in file:

tokens=line.split()

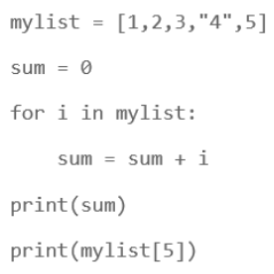
print("Wordcount:"+str(len(tokens))+" In line:",line)





**ASSIGNMENT 35**

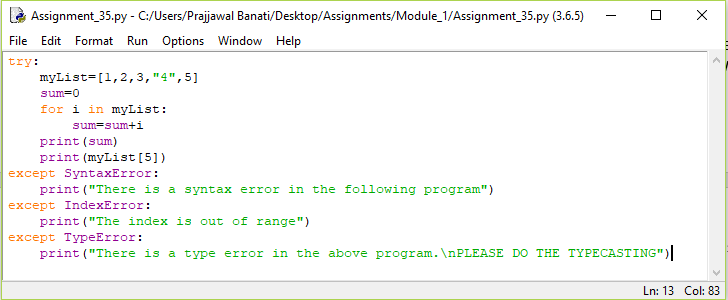
Assume the following Python code:

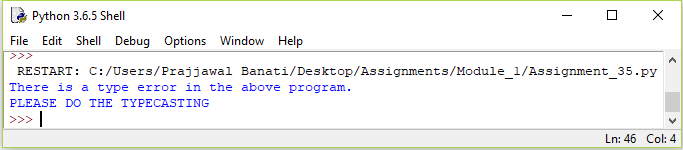


Rewrite the code to handle the exceptions raised. Print appropriate error messages wherever applicable.

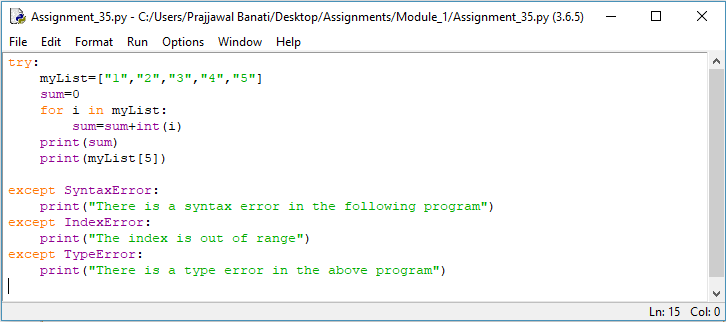
Ans: It is relevant from the above program that the type is not casted, Also the index is out of range

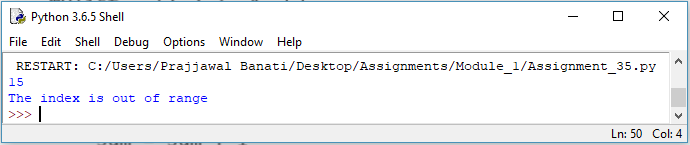
**SO INITIALLY THE PROGRAM WILL GIVE OUTPUT LIKE THIS:**



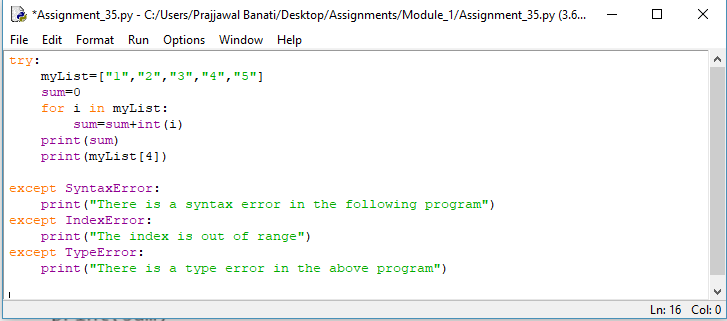


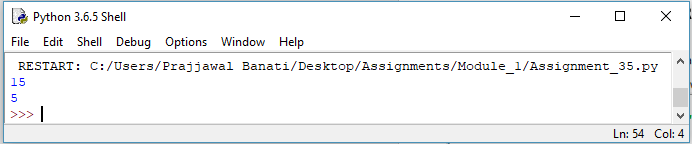
**NOW LET US REWRITE AND FIX THE TYPE ERROR:**





**SO NOW LETS FIX THE INDEX ERROR:**





**SO NOW THE PROGRAM IS CORRECT**

**ASSIGNEMENT 36**

You have already created a Python program to implement the following in ﬁle handling section:

1. read a ﬁle.

2. add backslash (\) before every double quote in the ﬁle contents.

3. write it to another ﬁle in the same folder.

4. print the contents of both the ﬁles.

Modify your code to implement Exception handling. Print appropriate error messages wherever applicable.

Ans:

try:

TestFile1=open("C:\\Users\\Prajjawal Banati\\Desktop\\Assignments\\Module\_1\\TestFile1.txt","w")

TestFile1.write('"Hello Pune"')

TestFile1.close()

TestFile1=open("C:\\Users\\Prajjawal Banati\\Desktop\\Assignments\\Module\_1\\TestFile1.txt","r+")

TestFile2=open("C:\\Users\\Prajjawal Banati\\Desktop\\Assignments\\Module\_1\\TestFile2.txt","w")

for line in TestFile1:

line.split(' ')

TestFile2.write('\\'+line+'\\')

TestFile2.close()

TestFile1.close()

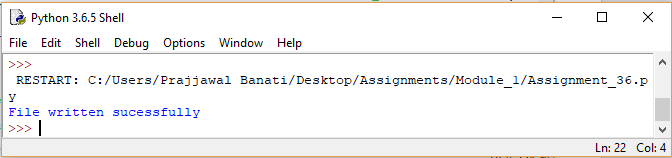
except IOError:

print("Please open the file in 'w' mode as the file does not exist")

else:

print("File written sucessfully")





**ASSIGNMENT 37**

You have already executed the Python program given below in Functions section:

• Add natural numbers up to n where n is taken as an input from user.

Do appropriate exception handling in the code and observe the output by providing invalid input values.

Ans:

print("Hello Everyone, This program will print the sum of all the numbers from 1 to n \

(User will input the value of n)")

n=int(input("Enter the value of n: "))

try:

def sum(n):

if n>=0:

print("So the sum of natural numbers from 1 to",n,"is: ")

k=0

for value in range(1,n+1):

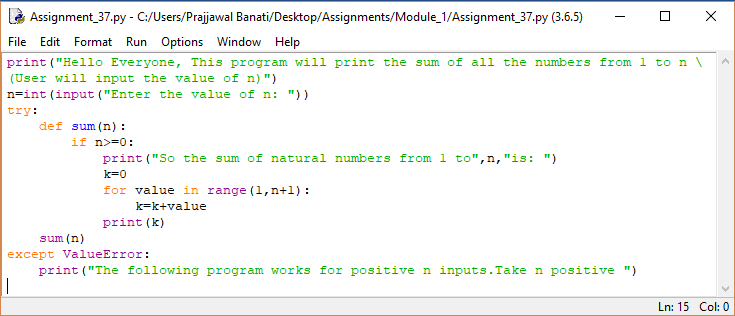
k=k+value

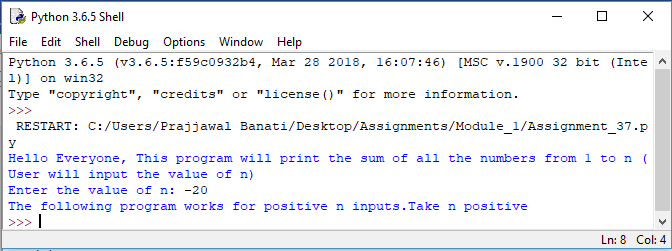
print(k)

sum(n)

except ValueError:

print("The following program works for positive n inputs.Take n positive ")





**ASSIGNMENT 38**

At an airport, a traveler is allowed entry into the ﬂight only if he clears the following checks:

1.Baggage Check

2.Immigration Check

3.Security Check

The logic for the check methods are given below:

check\_baggage (baggage\_weight)

•returns True if

baggage\_weight is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.

check\_immigration (expiry\_year)

•.returns True if expiry\_year is greater than or equal to 2001 and less than or equal to 2025. Otherwise returns False.

check\_security(noc\_status)

•.returns True if noc\_status is 'valid' or 'VALID', for all other values return False.

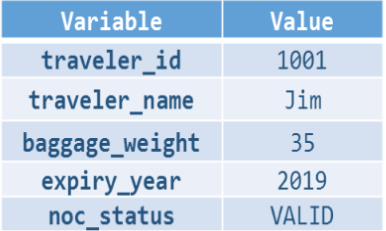
traveler()

•.Initialize the traveler Id and traveler name and invoke the functions check\_baggage(), check\_immigration() and check\_security() by passing required arguments.

•.Refer the table below for values of arguments.

•.If all values of check\_baggage(), check\_immigration() and check\_security() are true,

display traveler\_id and traveler\_name display "Allow Traveler to ﬂy!"



Otherwise,display traveler\_id and traveler\_name display "Detain Traveler for Re-checking!“ Invoke the traveler() function. Modify the values of diﬀerent variables in traveler() function and observe the output.

Ans:

print("Hello and welcome to the airport portal.To continue please enter the\

traveller\_id and traveler\_name")

def check\_baggage(baggage\_weight):

try:

if(baggage\_weight>=0 and baggage\_weight<=40):

return True

else:

return False

except:

print("Error occured in checking the baggage weight")

return None

def check\_immigration(expiry\_year):

try:

if(expiry\_year>=2001 and expiry\_year<=2025):

return True

else:

return False

except:

print("Error occured in checking the expiry year")

return None

def check\_security(noc\_status):

try:

if(noc\_status=="valid" or noc\_status=="VALID"):

return True

else:

return False

except:

print("Error occured in checking the status")

return None

def traveler():

traveler\_id=int(input("Enter the traveler\_id: "))

traveler\_name=input("Enter the traveler name: ")

baggage\_weight=int(input("Enter the baggage weight of the traveller: "))

a=check\_baggage(baggage\_weight)

expiry\_year=int(input("Enter the expiry year of the passport: "))

b=check\_immigration(expiry\_year)

noc\_status=input("Enter the status of noc: ")

c=check\_security(noc\_status)

if a==True and b==True and c==True:

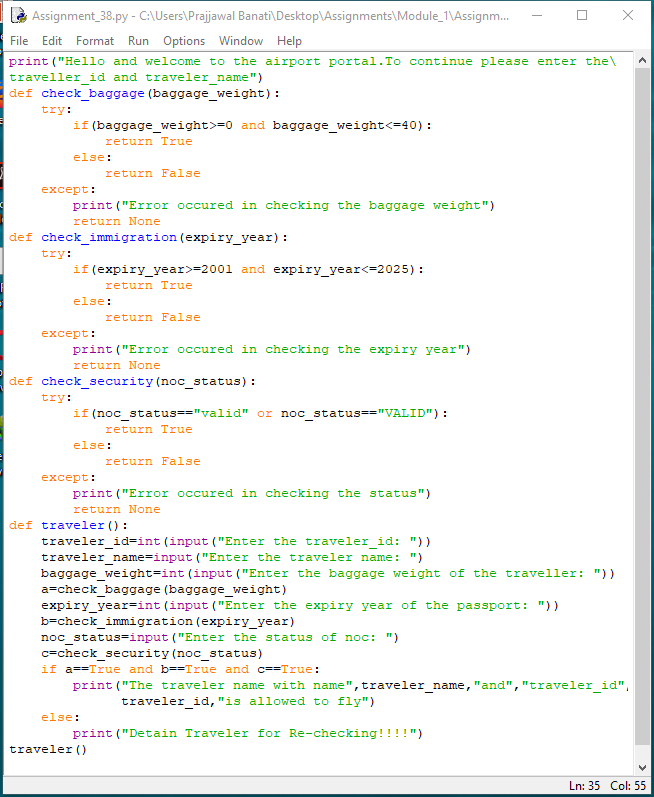
print("The traveler name with name",traveler\_name,"and","traveler\_id",

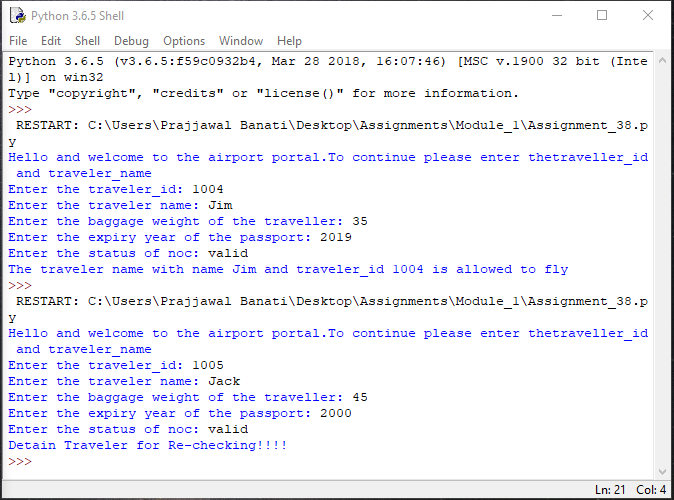
traveler\_id,"is allowed to fly")

else:

print("Detain Traveler for Re-checking!!!!")

traveler()





**ASSIGNMENT 39**

1. •Create a module "number\_checker.py" which has following 2 functions:

•is\_prime(num) : this function returns true if the input number is prime

•is\_even(num): this function returns true if the input number is even

•Create another Python module "test\_module.py".

•Invoke the functions "is\_prime(num)" and "is\_even(num)" in "test\_module.py".

•Observe the results.

Ans:

#number\_checker.py

def is\_prime(num):

k=0

for i in range(2,num):

if num%i==0:

k=1

if k==0:

return True

else:

return False

def is\_even(num):

k=0

if num%2==0:

return True

else:

return False

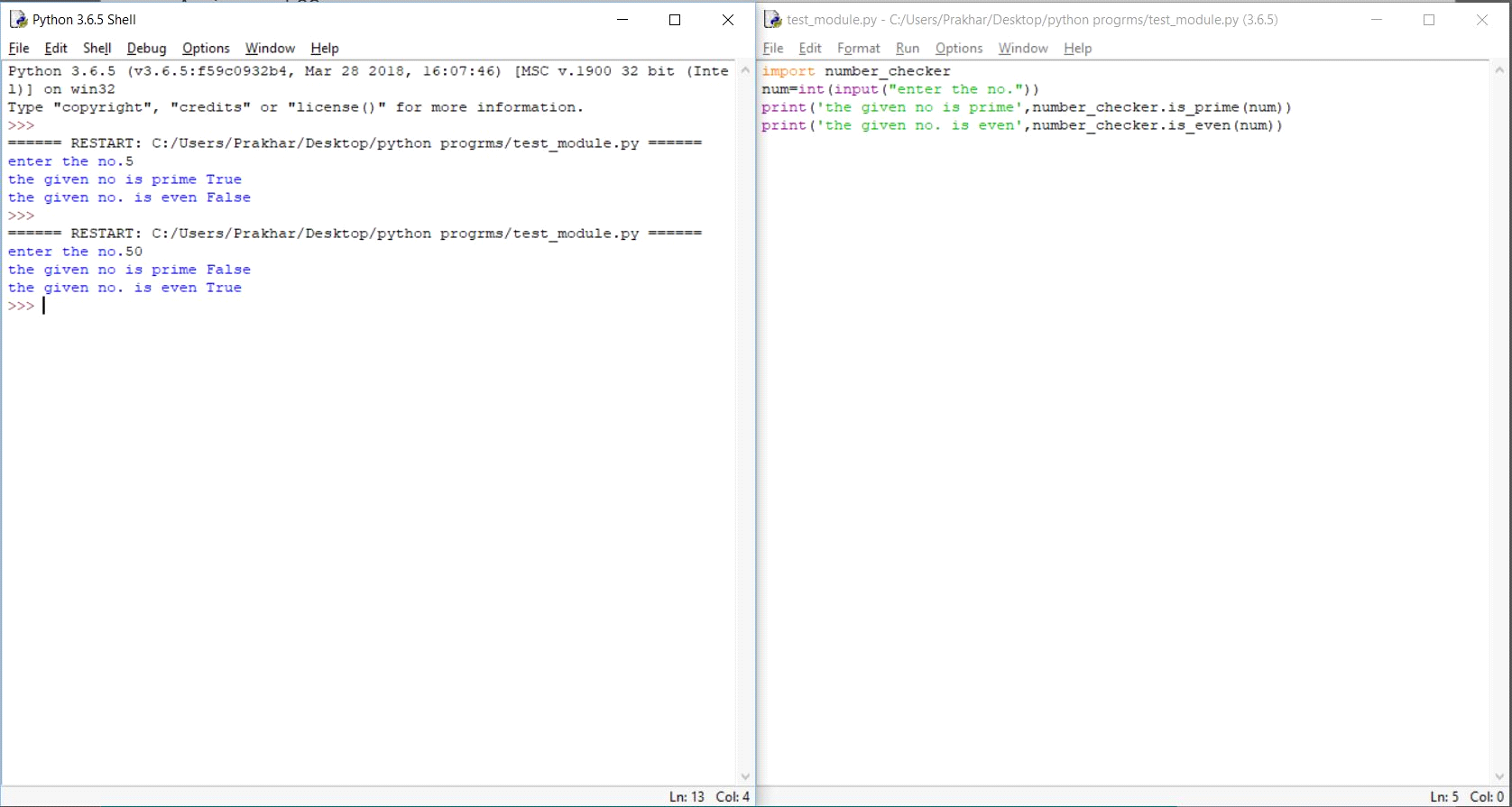
#test\_module

import number\_checker

num=int(input("enter the no."))

print('the given no is prime',number\_checker.is\_prime(num))

print('the given no. is even',number\_checker.is\_even(num))



**ASSIGNMENT 40**

1. Write a Python program to randomly print any of the below numbers:

100,200,300,400,500,600,700,800,900,1000

Execute the program 10 times and verify if the number generated in every output is one out of the numbers given in the list above.

Ans:

List=[]

import random

print("Randomly printed numbers are:\n")

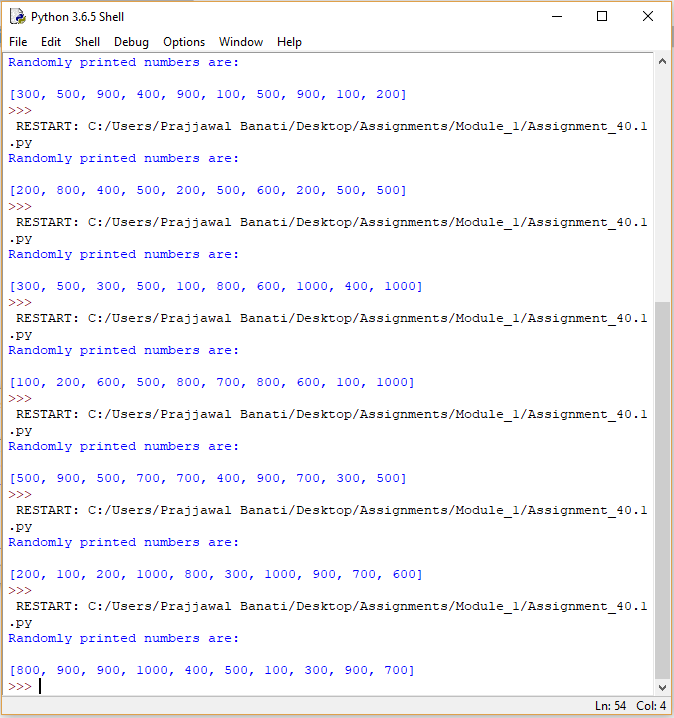
for i in range(1,11):

a=random.randint(1,10)\*100

List.insert(i,a)

print(List)





1. Write a Python program to print a random odd numbers between 10 and 50.

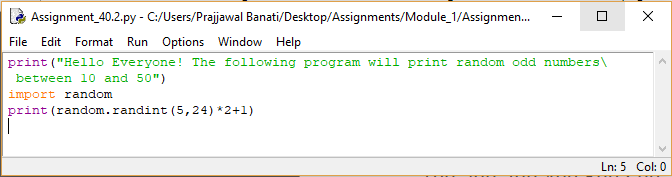
Ans:

print("Hello Everyone! The following program will print random odd numbers\

between 10 and 50")

import random

print(random.randint(5,24)\*2+1)





**ASSIGNMENT 41**

Write a Python program for rolling a dice on clicking enter key. The program should run inﬁnitely until user enters

'q'.

Ans:

print("Hello everyone! The following is a program of rolling dice")

q=0

import random

while q!='q':

num=random.randint(1,6)

if num==1:

print("Say the player to move by",num,"places")

print('Or press q to quit If want to continue press any key other than q')

q=input()

if num==2:

print("Say the player to move by",num,"places")

print('Or press q to quit If want to continue press any key other than q')

q=input()

if num==3:

print("Say the player to move by",num,"places")

print('Or press q to quit If want to continue press any key other than q')

q=input()

if num==4:

print("Say the player to move by",num,"places")

print('Or press q to quit If want to continue press any key other than q')

q=input()

if num==5:

print("Say the player to move by",num,"places")

print('Or press q to quit If want to continue press any key other than q')

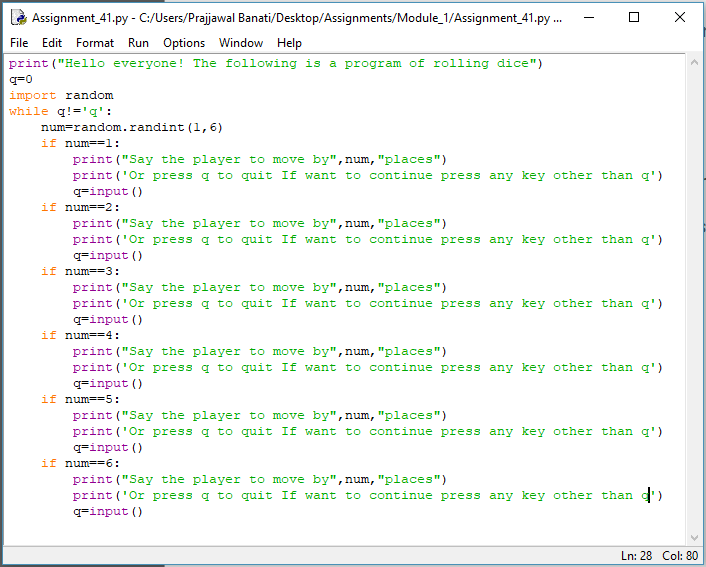
q=input()

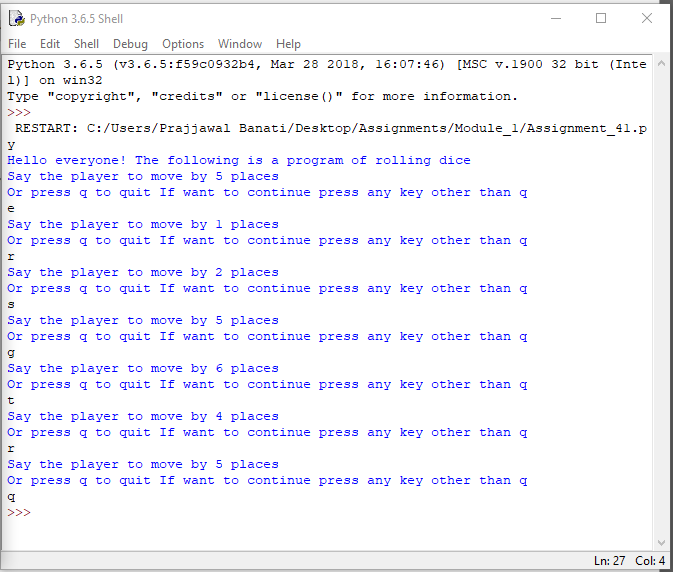
if num==6:

print("Say the player to move by",num,"places")

print('Or press q to quit If want to continue press any key other than q')

q=input()





**ASSIGNMENT 42**

If area of one wall of a cubical wooden box is 16 units, write a Python program to display the volume of the box.

Note:

Area of a cube with side 'a' is 'a\*\*2'.

Volume of the cube can be computed as 'a\*\*3'.

Ans:

print("If area of one wall of a cubical wooden box is 16 units, \

write a Python program to display the volume of the box. ")

import math

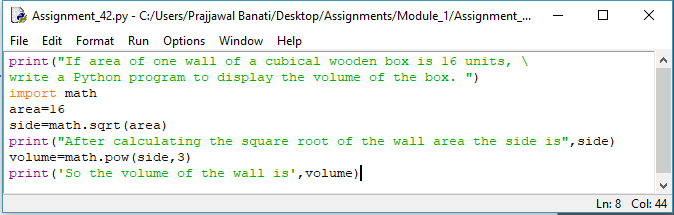
area=16

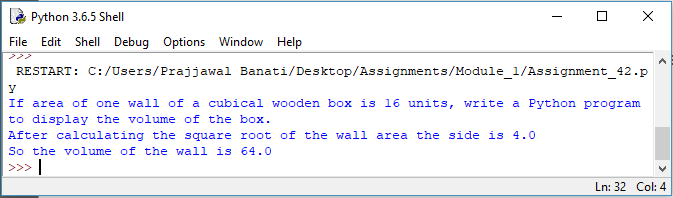
side=math.sqrt(area)

print("After calculating the square root of the wall area the side is",side)

volume=math.pow(side,3)

print('So the volume of the wall is',volume)





**ASSIGNMENT 43**

The ABC Institute oﬀers vocational courses to students in multiple areas e.g. theatre, classical singing, traditional dance forms, Bollywood dance, literature and so on. A student can enroll for zero to all courses.

Write a Python function that takes the number of courses as an input and returns the total number of diﬀerent course combinations, a student can opt for. (Make use of functions available in math module.

Ans:

print("Hello everyone! Welcome The ABC Institute oﬀers vocational courses \

to students in multiple areas e.g. theatre, classical singing, traditional dance \

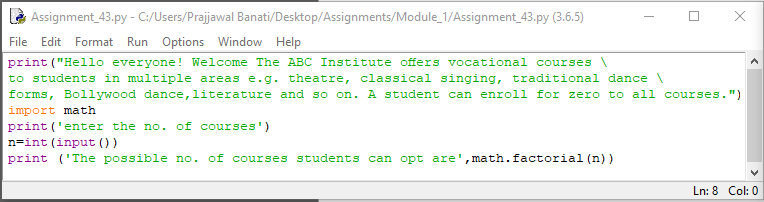
forms, Bollywood dance,literature and so on. A student can enroll for zero to all courses.")

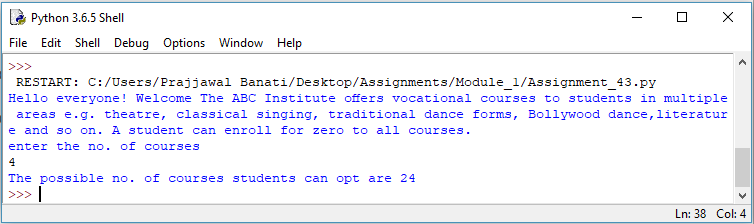
import math

print('enter the no. of courses')

n=int(input())

print ('The possible no. of courses students can opt are',math.factorial(n))





**ASSIGNMENT 44**

Execute the following code and observe the output.

1. import time

2. print(time.time())

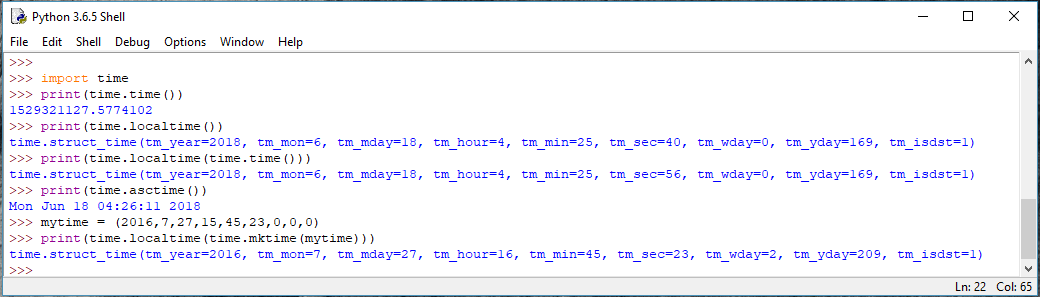
3. print(time.localtime())

4. print(time.localtime(time.time()))

5. print(time.asctime())

6. mytime = (2016,7,27,15,45,23,0,0,0)

7. print(time.localtime(time.mktime(mytime)))



**ASSIGNMENT 45**

Consider a Python string:

cust\_details = "Hello John, your customer id is j181"

1) Find, if the name of the customer is preceded by a pattern "Hello " or "hello " (Observe a space after the word)? If pattern is found, print the searched result.

2) Find, if the given string ends with a pattern containing only one alphabet followed by three numbers? If pattern is found, print the searched result.

3) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).

4) Replace the word "id" with "ID".The output of the above code is "Hello John, your customer ID is 181

Ans:

cust\_details="Hello John, your customer id is j181"

import re

print("1 Find if the customer name is preceded by hello or Hello and mention if yes")

match=re.search("hello |Hello ",cust\_details,re.I)

if (match.group()!=None):

print("The name is preceded by Hello or hello as:",match.group())

else:

print("Name is not greeted as asked")

print("")

print("")

print("2) Find, if the given string ends with a pattern containing only one alphabet followed by three numbers? If pattern is found, print the searched result")

match=re.search("\D\d{3}",cust\_details)

if match.group()!=None:

print("The query is true and has result:",match.group())

else:

print("Query is not true")

print("")

print("")

print('''3) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).''')

cust\_details=re.sub("j181","181",cust\_details)

print(cust\_details)

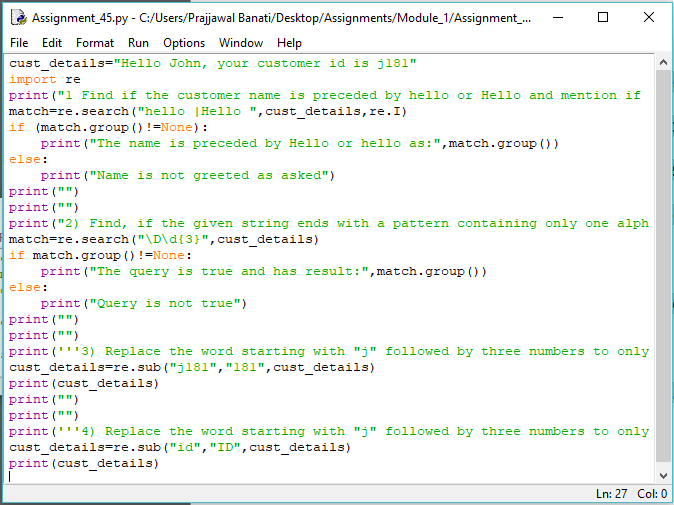
print("")

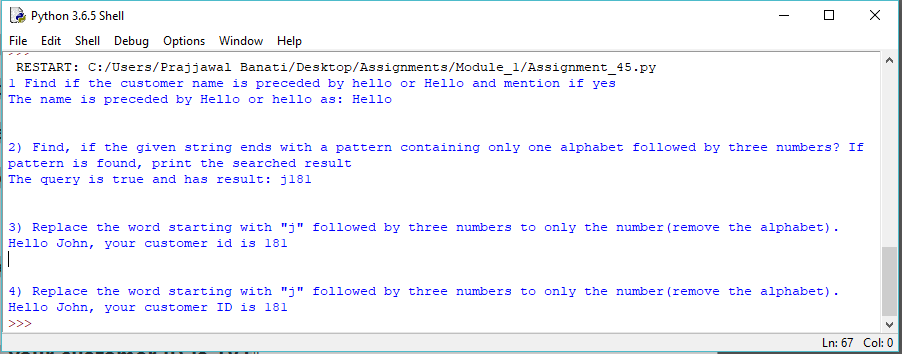
print("")

print('''4) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).''')

cust\_details=re.sub("id","ID",cust\_details)

print(cust\_details)





**ASSIGNMENT 46**

Consider a scenario of managing student details in ABC Training Institute. Write a Python program to implement the

business requirements mentioned below:

a)Accept student\_id and validate whether it contains only digits.

b)If student \_id is valid, accept student\_name from the user and validate whether it contains only alphabets.

c)If student\_name is valid, accept fees\_amount paid by the student:

1.Decimal point is optional in fees\_amount(can have maximum one decimal point)

2.Only two digits are allowed after decimal point

d)If invalid data is entered in any of the above steps, display appropriate error messages. Else, create an email\_id for student as

student\_name@ABC.com

. Assume there are no duplicate names.

e)Perform above validations using Regular Expressions and print details of the student: student\_id, student\_name, fees\_amount, email\_id

Ans:

institute="ABC Training Institute"

student\_id=(input("Enter the student ID: "))

import re

email\_id=str() #empty string declaration

if re.search("^\d+$",student\_id):

student\_name=input("Enter name of the Student: ")

if re.search("^[a-zA-Z]+$",student\_name): #Question says only alphabets and thus space is not allowed in check so name can be only one word

fees\_amount=input("Enter the fees to be paid as a number with atmost two precision after decimal: ")

if re.search("^\d+.\d{2}",fees\_amount):

if fees\_amount.count(".")>1:

print("An amount cannot have more than one period symbol")

else:

email\_id=student\_name+"@ABC.com"

else:

print("Wrong Format for fees")

else:

print("Name must have only alphabets in it!!!")

else:

print("Enter student id only as combination of digits!!")

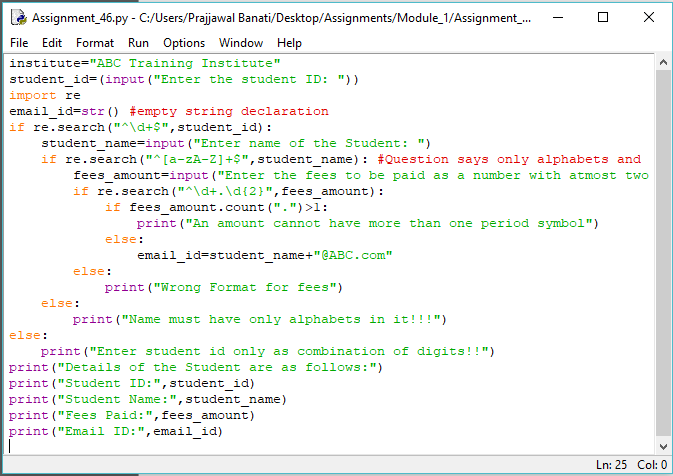
print("Details of the Student are as follows:")

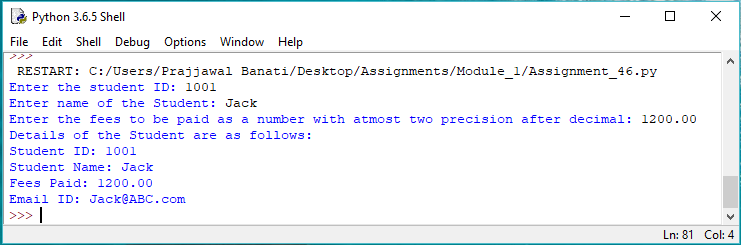
print("Student ID:",student\_id)

print("Student Name:",student\_name)

print("Fees Paid:",fees\_amount)

print("Email ID:",email\_id)





**ASSIGNMENT 47**

Consider a string:

my\_string = “””Strings are amongst the most popular data types in Python. We can create the strings by enclosing

characters in quotes. Python treats single quotes the same as double quotes.”””

1) Write a Python program to count the number of occurrences of word "String" in the given string ignoring the

case.

2) Write a function "count\_words" to print the count of occurrences of a word:

a) which end with "on". (e.g. Python)

b) which have "on" in between the ﬁrst and last characters (e.g. amongst)

Ans:

my\_string="""Strings are amongst the most popular data types in Python. We can create the strings by enclosing \

characters in quotes. Python treats single quotes the same as double quotes."""

print("The number of occurences of word string are:",my\_string.lower().count("string"))

def count\_words(str1):

list1=str1.split()

for k in range(0,len(list1)):

if list1[k].endswith("on")==True:

print("Occurences of word",list1[k],"with on at end are:",str1.count(list1[k]))

else:

continue

for k in range(0,len(list1)):

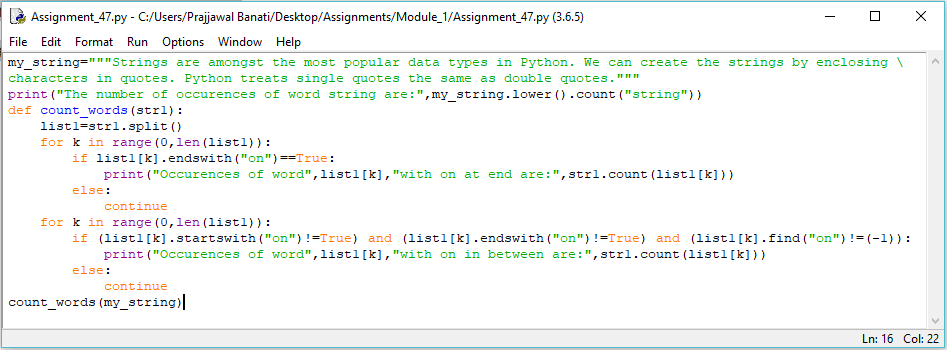
if (list1[k].startswith("on")!=True) and (list1[k].endswith("on")!=True) and (list1[k].find("on")!=(-1)):

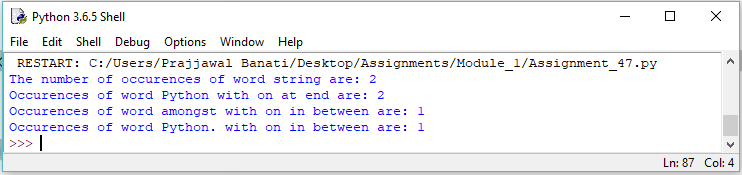
print("Occurences of word",list1[k],"with on in between are:",str1.count(list1[k]))

else:

continue

count\_words(my\_string)





**ASSIGNMENT 48**

Consider the price list of various items in the Retail Store:

item\_price = [1050, 2200, 8575, 485, 234, 150, 399]

Customer John wants to know the:

1. Price of costliest item sold in retail store

2. Number of items in the Retail store

3. Prices of items in increasing order

4. Prices of items in descending order

Implement the above mentioned business requirements using built-in List functions.

Ans:

print("Hello Everyone! Welcome to Retail store. The follwing are the stats of items \

we sell")

print("Retail Store Prices")

item\_price=[1050,2200,8575,485,234,150,399]

customer\_name=input("Enter the customer name: ")

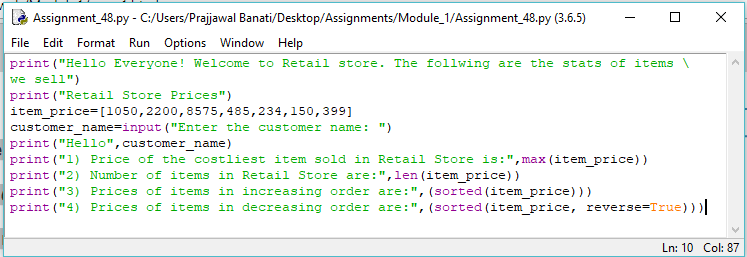
print("Hello",customer\_name)

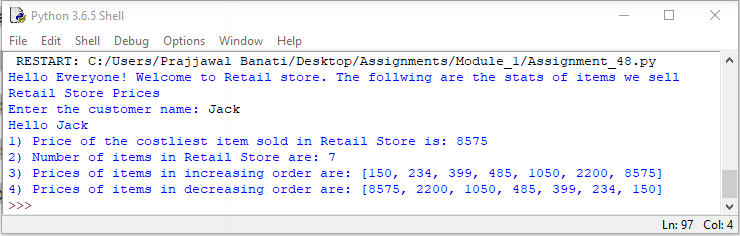
print("1) Price of the costliest item sold in Retail Store is:",max(item\_price))

print("2) Number of items in Retail Store are:",len(item\_price))

print("3) Prices of items in increasing order are:",(sorted(item\_price)))

print("4) Prices of items in decreasing order are:",(sorted(item\_price, reverse=True)))





**ASSIGNMENT 49**

Built-in function in Python which accepts any data structure (list, string, tuple, dictionary and set) and returns a

sorted list.

Ans:

student\_tuples=[('John','A',15),('Jane','B',12),('Dave','B',10)]

print(sorted(student\_tuples,key=lambda student : student[2]))

print(sorted(student\_tuples,key=lambda student : student[2],reverse=True))

