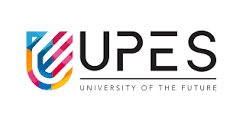
**RECORD OF EXPERIMENT**

Python Programming Lab

**MASTER OF COMPUTER APPLICATION**

TO



**UNIVERSITY OF PERTOLEUM AND ENERGY STUDIES**

BY

**ANKESH MOBAR**

**SAP ID: 500100953**

**Enroll no.: R271222025**

UNDER THE GUIDENCE OF

**PROF. SHELLY A**

**SCHOOL OF COMPUTER SCIENCE**

**UNIVERSITY OF PERTOLEUM AND ENERGY STUDIES**

**2022-2023**

## ACKNOWLEDGEMENTS

It is indeed a great pleasure to express our sincere thanks to our august supervisor, **PROF. SHELLY A** school of computer science of UNIVERSITY OF PETROLEUM AND ENERGY STUIDIES, Dehradun for his continuous support in this project. He was always there to listen and to give advice. He showed us different ways to approach a research problem and the need to be persistent to accomplish any goal. He taught us how to write academic paper, had confidence in us when we doubted ourselves, and brought out the good ideas in us. He was always there to meet and talk about our ideas, to proofread and mark up our paper, and to ask us good questions to help us think through our problems. Without his encouragement and constant guidance, we could not have finished this project.

We are thankful to our family whose unfailing love, affection, sincere prayers and best wishes had been a constant source of strength and encouragement.

Last, but not least, we thank our parents, for giving us life in the first place, for educating us with aspects from both arts and sciences, for unconditional support and encouragement to pursue our interests. We dedicate this work to our parents who will feel very proud of us. They deserve real credit for getting us this far, and no words can ever repay for them.

**ANKESH MOBAR**

**MCA**

**SAPID: 500100953**

**Enroll no.: R271222025**

**CONTENT**

* **Assignment 1………………………………….** page no: 4
* **Assignment 2………………………………….** page no:9
* **Assignment 3………………………………….** page no:12
* **Assignment 4………………………………….** page no:18
* **Assignment 5………………………………….** page no:21
* **Assignment 6………………………………….** page no:24
* **Assignment 7………………………………….** page no:27
* **Assignment 8………………………………….** page no:34
* **Assignment 9………………………………….** page no:37

**Assignment: 1**

**Question. Why Python?**

**Answer:** Python is one of the most loved programming languages by developers, data scientists, software engineers, and even hackers because of its versatility, flexibility, and object-oriented features.   
Although it's a high-level language and but still we can do variety of complex tasks through It is easy to learn and has a clean syntax.

**Question. what are prev versions of python and differences?**

**Answer:**

**Python 3.8:** In rasa installation docs it says rasa works with python 3.8, but unfortunately this demo throws the following error while installing pip dependencies.

**Python 3.9:** Python 3.9 is the last version providing those Python 2 backward compatibility layers, to give more time to Python projects maintainers to organize the removal of the Python 2 support and add support for Python 3.9.

**Python 3.10:** Aliases to [Abstract Base Classes](https://docs.python.org/3/library/collections.abc.html#collections-abstract-base-classes) in the [collections](https://docs.python.org/3/library/collections.html#module-collections) module, like collections. Mapping alias to [collections.abc.Mapping](https://docs.python.org/3/library/collections.abc.html#collections.abc.Mapping), are kept for one last release for backward compatibility. They will be removed from Python 3.10.

**Question: Basic command of python.**

**Answer:** input, print, range, round, pip install, len, sort, loop, %run, %cd.

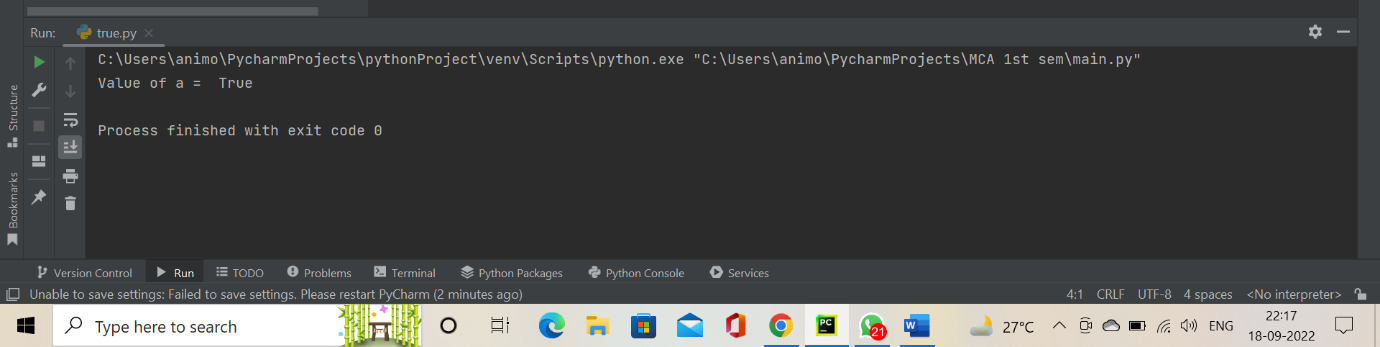
**Experiments**

1. Evaluate following and express the results:
2. 5 > 2.

Coding:

a = 5 > 2  
print("Value of a = ",a)

Output:



1. 5 == 2.

Coding:

a = 5==2  
print("Value of a = ",a)

Output:

1. 5 = 2.

Coding:

a = 5 = 2  
print("Value of a = ",a)

Output:

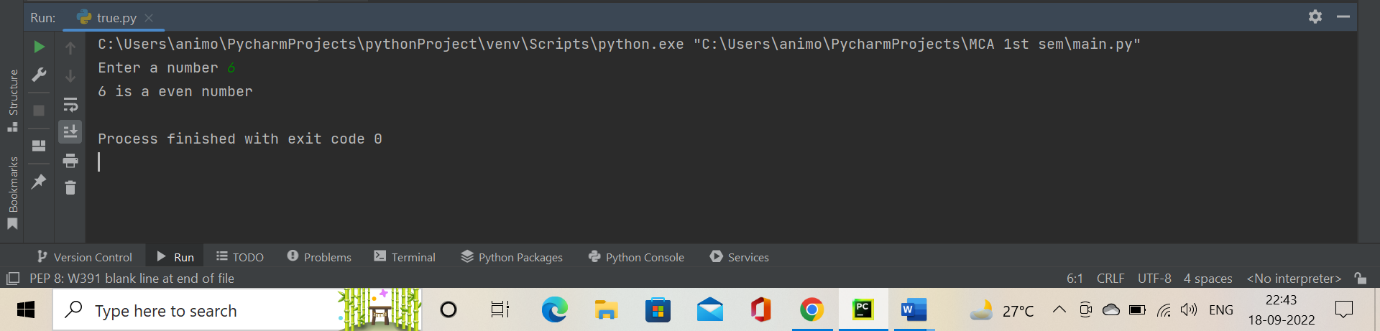


1. Determine whether the given number is even or odd.

Coding:

number=(int(input("Enter a number ")))  
  
if (number % 2) == 0:  
 print("{0} is a even number".format(number))  
else:  
 print("{0} is a odd number".format(number))

Output:

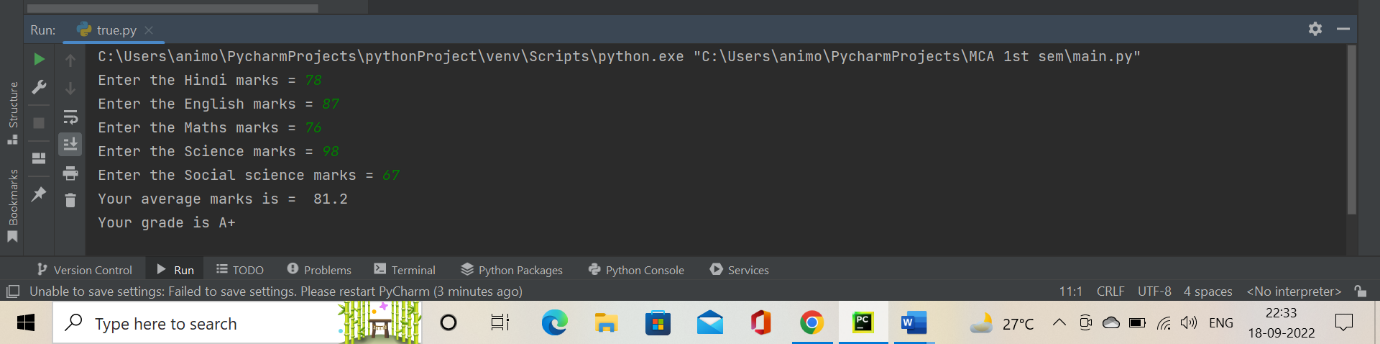


1. Marks of a student in five subjective are acquired. Determine average marks and grade. O - 91-100 A+ 81 -90 A 71 - 80 B+ 61-70 B 51 – 60 C+ 41 - 50 C 35 - 40 Fails otherwise.

Coding:

number1 = (int(input("Enter the Hindi marks = " )))  
number2 = (int(input("Enter the English marks = " )))  
number3 = (int(input("Enter the Maths marks = " )))  
number4 = (int(input("Enter the Science marks = " )))  
number5 = (int(input("Enter the Social science marks = " )))  
AVERAGE = (number1 + number2 + number3 + number4 + number5)/5;  
print("Your average marks is = ",AVERAGE)  
if AVERAGE<=100 and AVERAGE>=91:  
 print("Your grade is O")  
elif(AVERAGE<=90) and (AVERAGE>=81):  
 print("Your grade is A+")  
elif(AVERAGE<=80) and (AVERAGE>=71):  
 print("Your grade is A")  
elif(AVERAGE<=70) and (AVERAGE>=61):  
 print("Your grade is B+")  
elif(AVERAGE<=60) and (AVERAGE>=51):  
 print("Your grade is B")  
elif(AVERAGE<=50) and (AVERAGE>=41):  
 print("Your grade is C+")  
elif(AVERAGE<=40) and (AVERAGE>=35):  
 print("Your grade is C")  
else:  
 print("Sorry your are fail...")

Output:

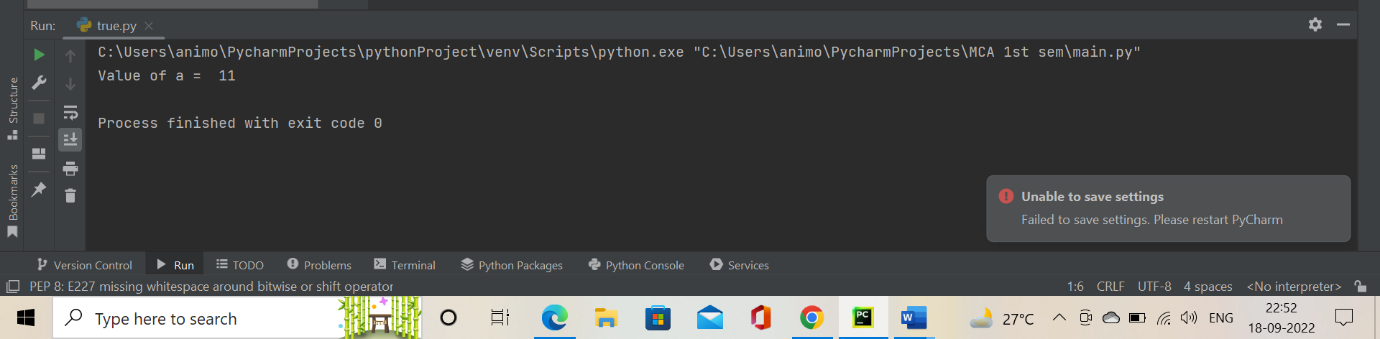


1. Evaluate 15 & 27 and verify your result.

Coding:

a=15&27  
print("Value of a = ",a)

Output:

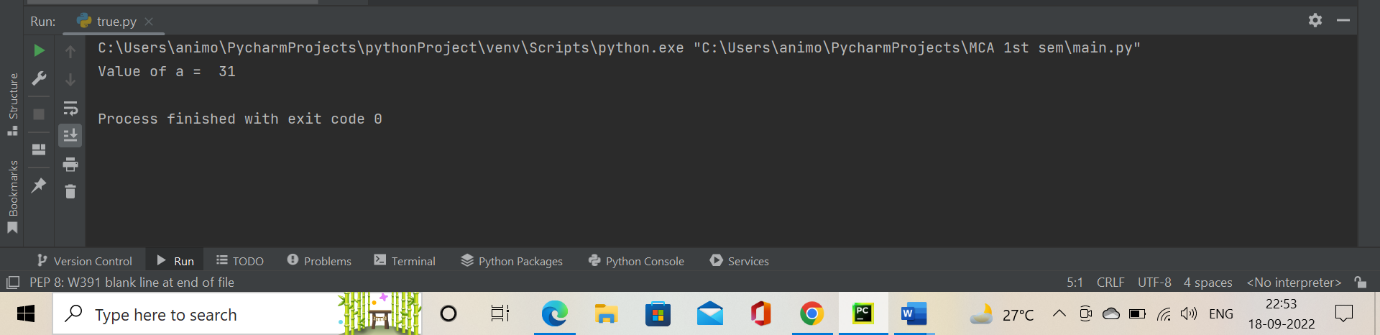


1. Evaluate 15 | 27 and verify your result.

Coding:

a=15|27  
print("Value of a = ",a)

Output:

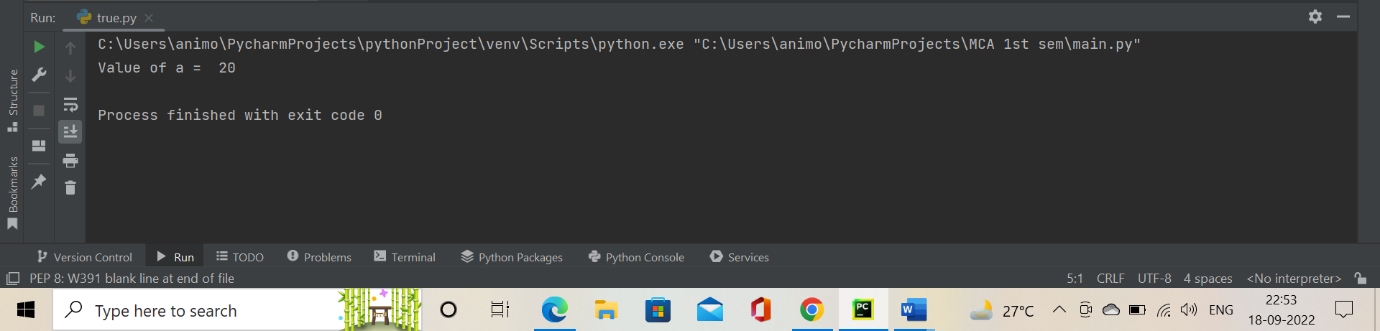


1. Evaluate 15 ^ 27 and verify your result.

Coding:

a=15^27  
print("Value of a = ",a)

Output:

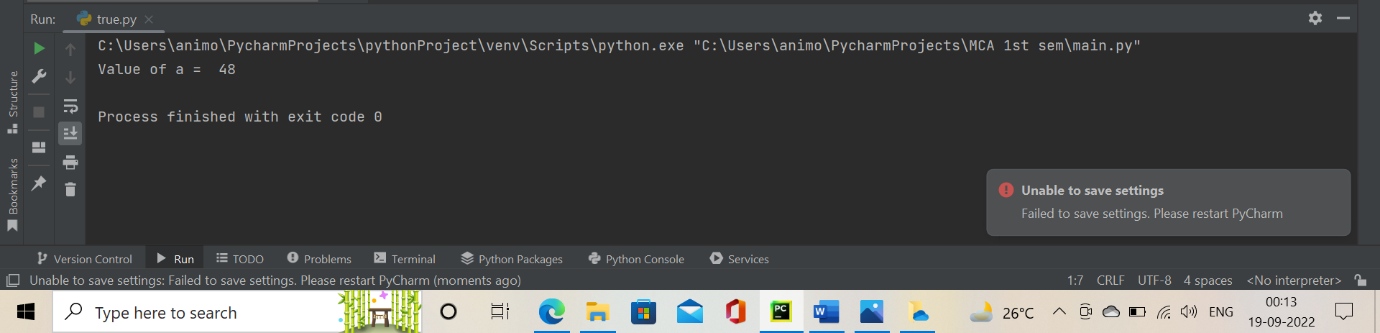


1. Multiply 3 by 4 using shift bitwise operator.

Coding:

a=3<<4  
print("Value of a = ",a)

Output:

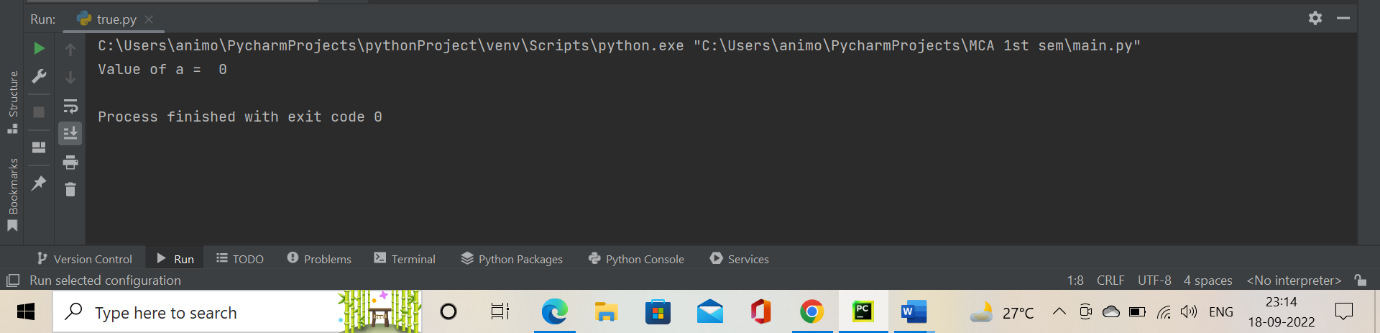


1. Divide 64 by 8 using shift bitwise operator.

Coding:

a=64>>8  
print("Value of a = ",a)

Output:



**Assignment: 2**

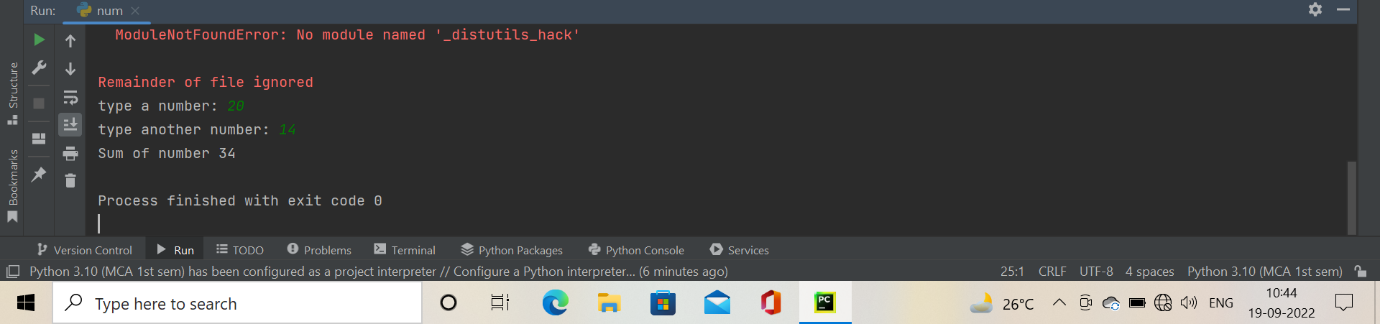
**Experiments**

1. **WAP add to numbers.**

**Coding:**

num1= int(input("type a number: "))  
num2= int(input("type another number: "))  
sum=num1+num2  
print('Sum of number',sum)

**Output:**

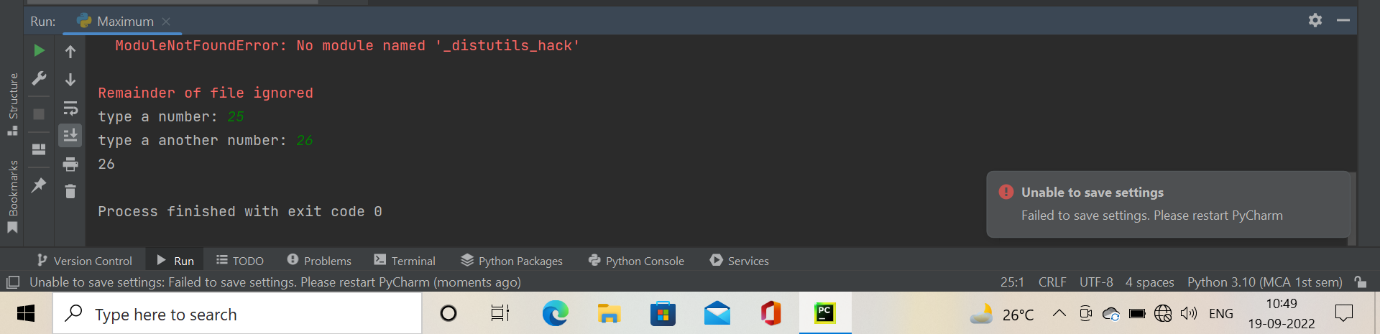


1. **WAP maximum of 2 numbers.**

**Coding:**

a= int(input("type a number: "))  
b= int(input("type a another number: "))  
print(a if a>b else b)

**Output:**

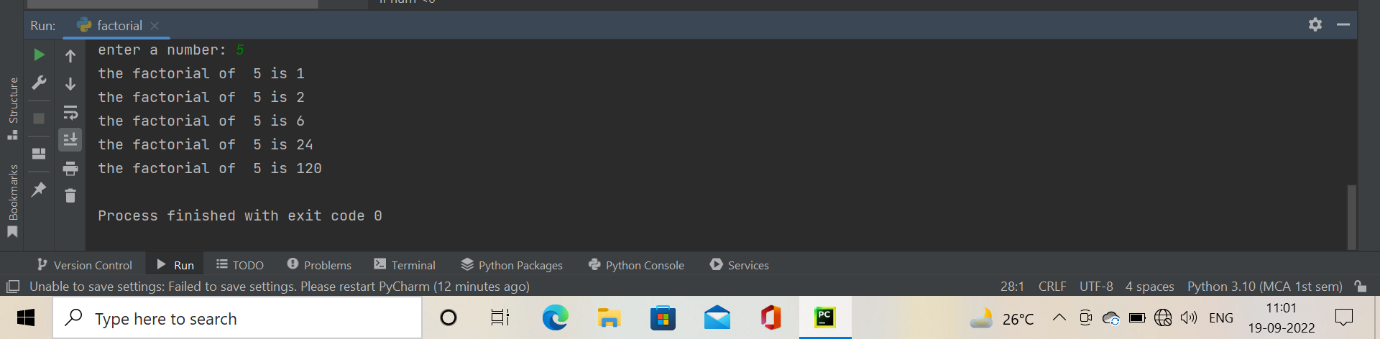


1. **WAP find factorial of a number.**

**Coding:**

num=int(input("enter a number: "))  
fact=1  
if num <0:  
 print("sorry. factorial does not exist for -ve number")  
elif num==0:  
 print("the factorial of 0 is 1 ")  
else:  
 for i in range(1,num+1):  
 fact=fact\*i  
 print("the factorial of ", num,"is",fact)

**Output:**

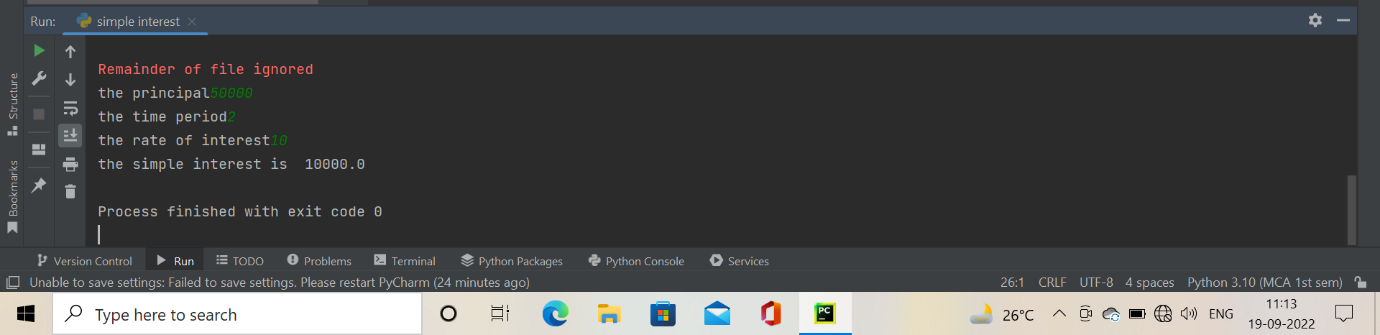


1. **WAP find simple interest.**

**Coding:**

p=int(input("the principal"))  
t=int(input("the time period"))  
r=int(input("the rate of interest"))  
si=(p\*t\*r)/100  
print('the simple interest is ',si)

**Output:**

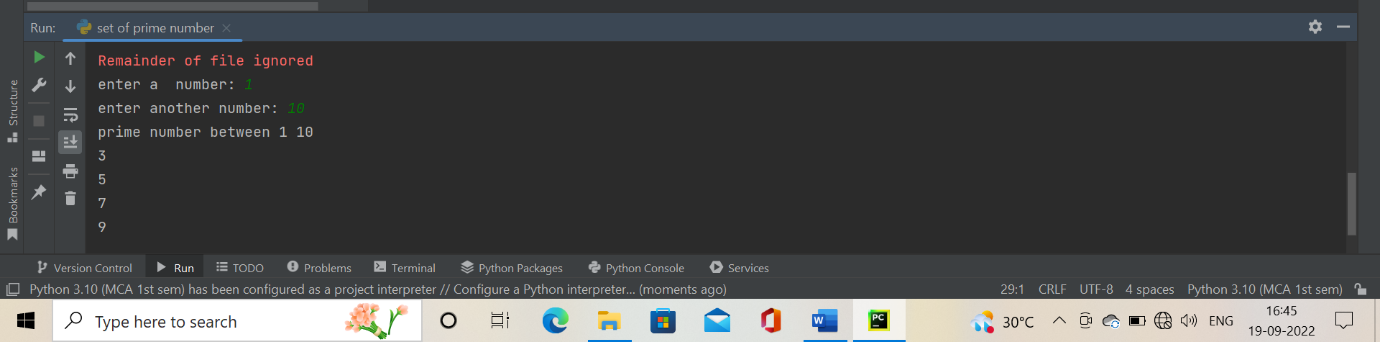


1. **WAP find prime number set.**

**Coding:**

a=int(input("enter a number: "))  
b=int(input("enter another number: "))  
print("prime number between",a,b)  
for num in range (a,b+1):  
 if num>1:  
 for i in range (2,num):  
 if (num%i==0):  
 break  
 else:  
 print(num)  
 break

**Output:**

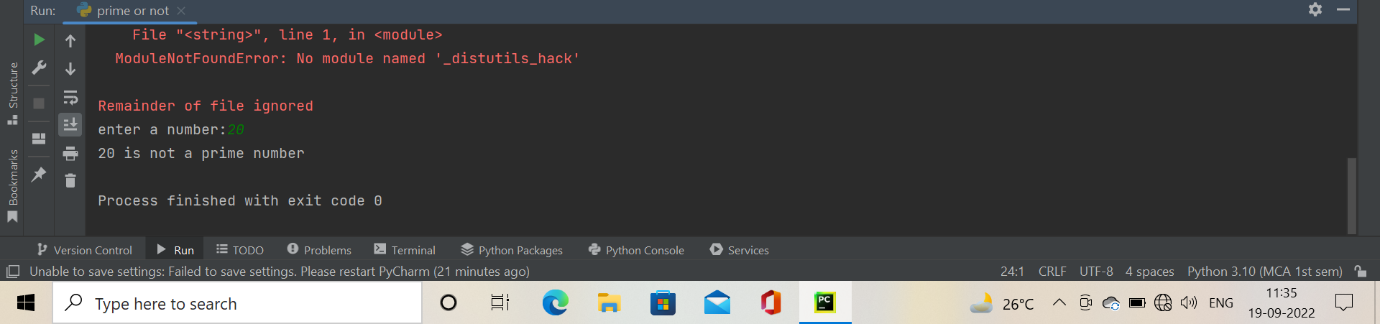


1. **WAP to check if a number is prime or not.**

**Coding:**

num=int(input("enter a number:"))  
flag = False  
if num>1:  
 for i in range(2,num):  
 if (num%i==0):  
 flag= True  
 break  
 if flag:  
 print(num,"is not a prime number")  
 else:  
 print(num,"is a prime number")

**Output:**



**Assignment: 3**

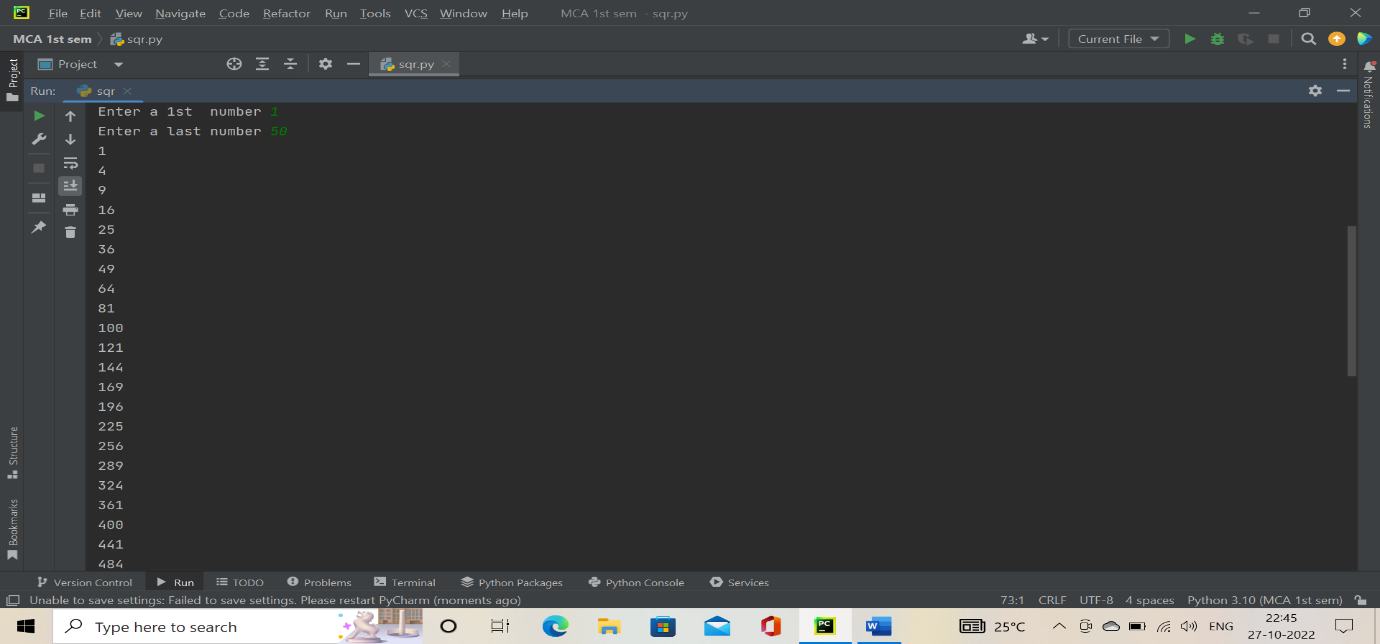
**Experiments**

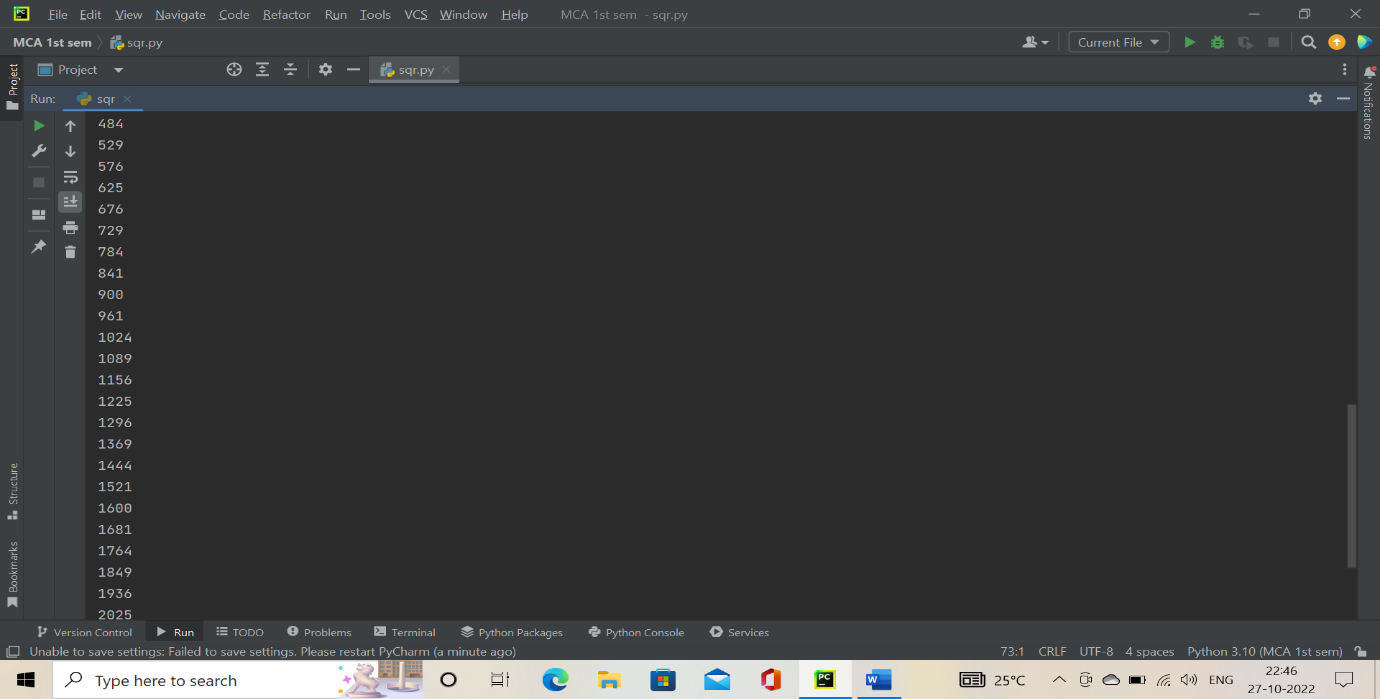
1. **Generate squares of all the integers from 1 to 50.**

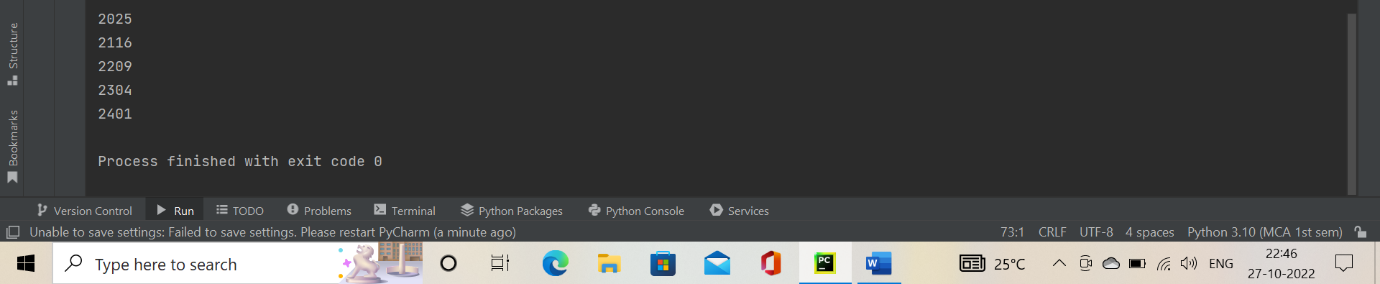
**Coding:**

n1=int(input("Enter a 1st number"))  
n2=int(input("Enter a last number"))  
def printValues():  
 sqr = list()  
 for i in range(n1,n2):  
 sqr=(i\*i)  
 print(sqr)  
printValues()

**Output:**





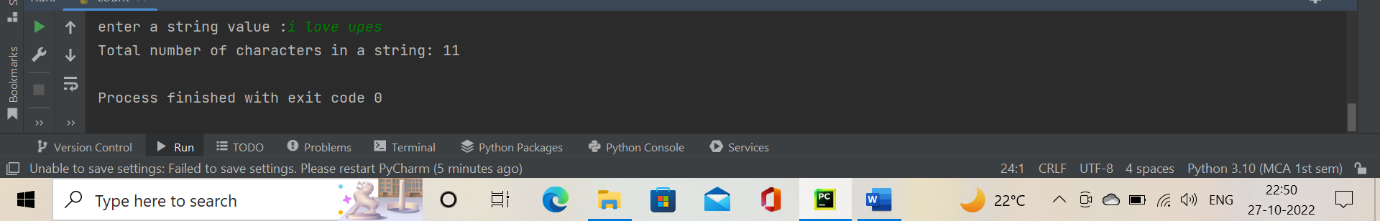


1. **Count the number of characters in a string using a loop.**

**Coding:**

string = input("enter a string value :")  
count = 0  
for i in range(0, len(string)):  
 if [(string[i] != ' ')]:  
 count = count + 1  
print("Total number of characters in a string: " + str(count))

**Output:**

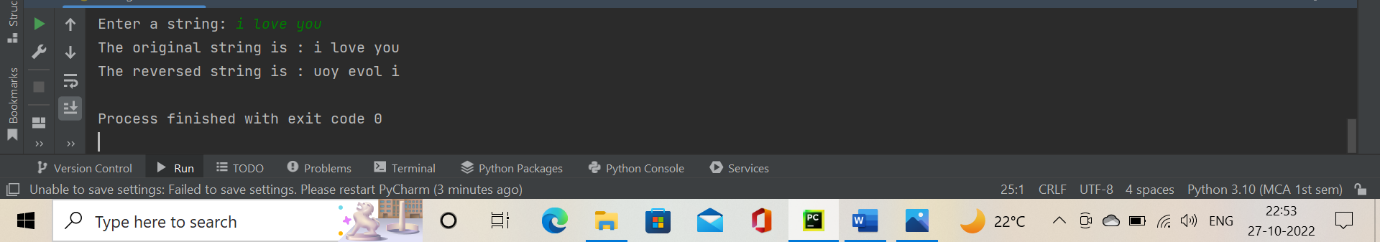


1. **Print a string in reverse.**

**Coding:**

def reverse(string):  
 string = string[::-1]  
 return string  
s=input("Enter a string: ")  
print("The original string is : ", end="")  
print(s)  
print("The reversed string is : ", end="")  
print(reverse(s))

**Output:**

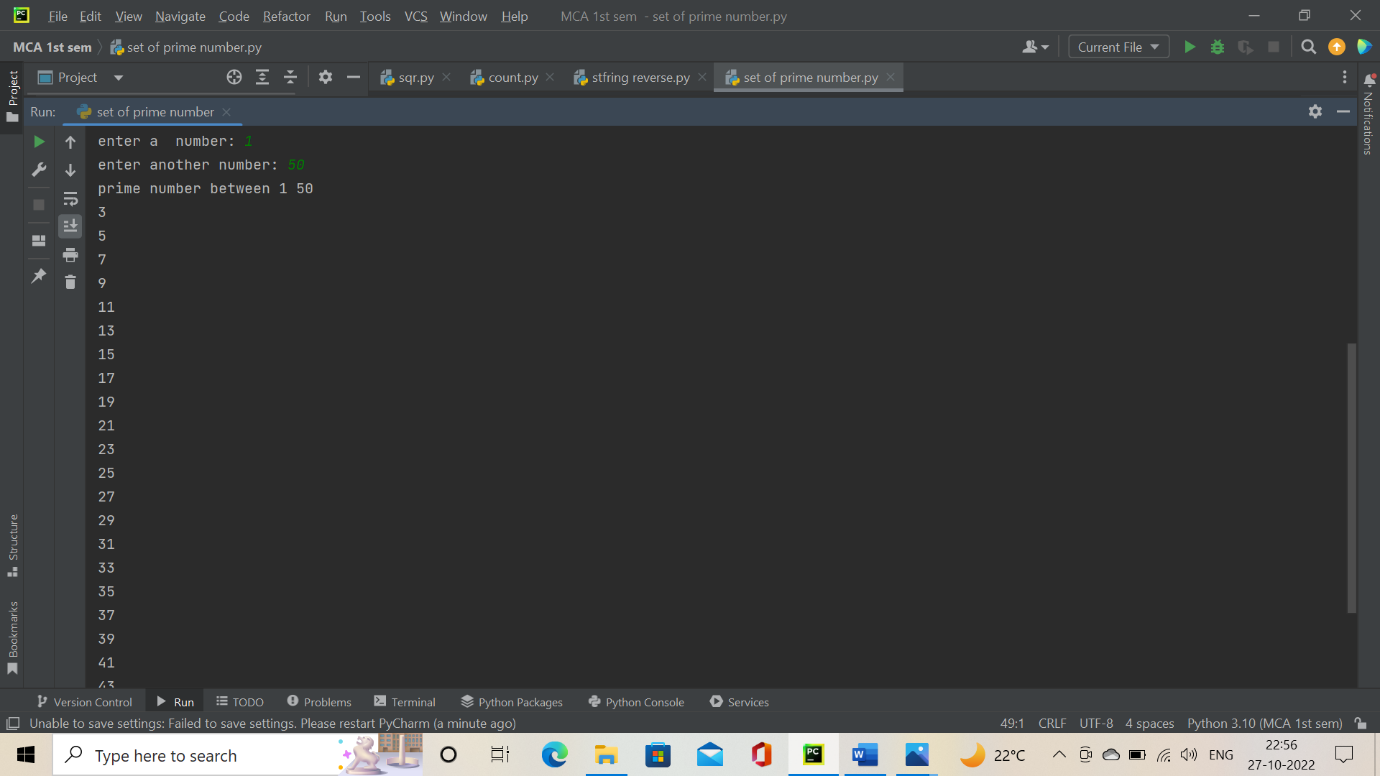


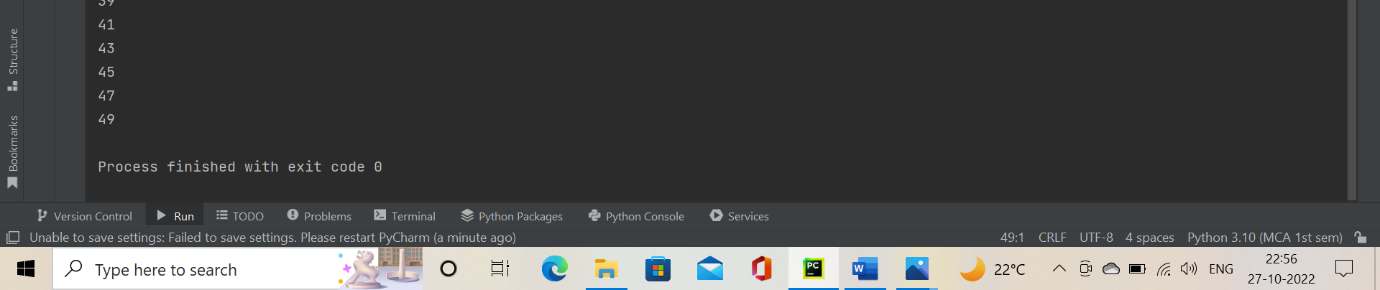
1. **Find all the prime numbers below 50.**

**Coding:**

a=int(input("enter a number: "))  
b=int(input("enter another number: "))  
print("prime number between",a,b)  
for num in range (a,b+1):  
 if num>1:  
 for i in range (2,num):  
 if (num%i==0):  
 break  
 else:  
 print(num)  
 break

**Output:**



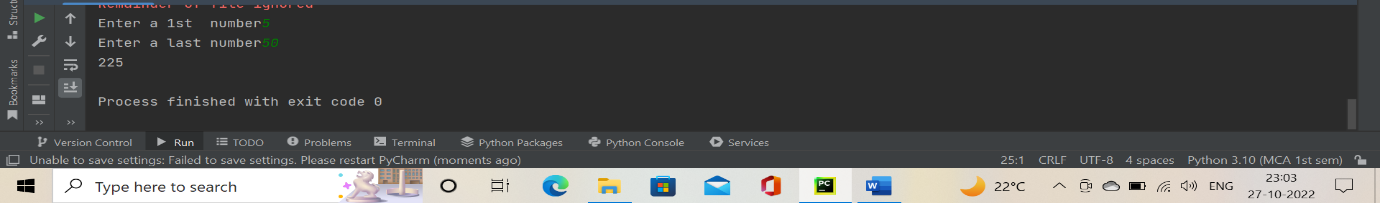


1. **Sum all the multiple integers of 5 below 50.**

**Coding:**

n1=int(input("Enter a 1st number"))  
n2=int(input("Enter a last number"))  
n = 0  
for i in range(n1,n2):  
 if not i % 5:  
 n = n + i  
print(n)

**Output:**



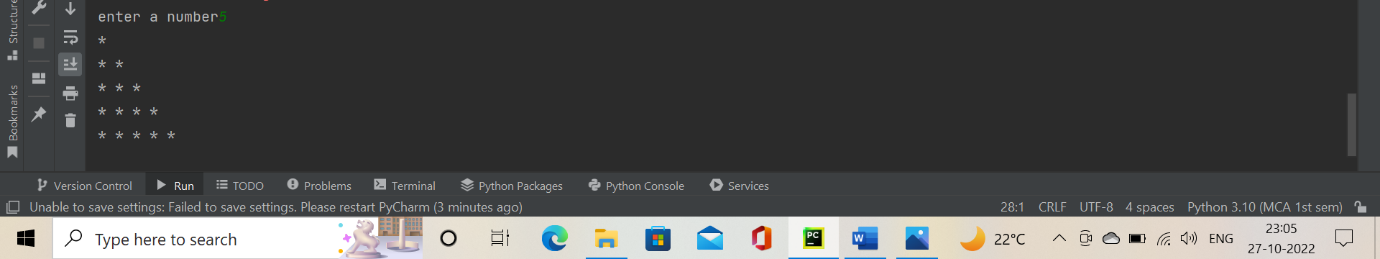
1. **Generate the patterns given below:**

|  |  |  |
| --- | --- | --- |
| **\***  **\* \***  **\* \* \* \***  **\* \* \* \* \*** | **1**  **2 1 2**  **3 2 1 2 3**  **4 3 2 1 2 3 4** | **1 2 3 4 5**  **2 3 4 5**  **3 4 5**  **4 5**  **5** |
|  |  |  |

**Coding:**

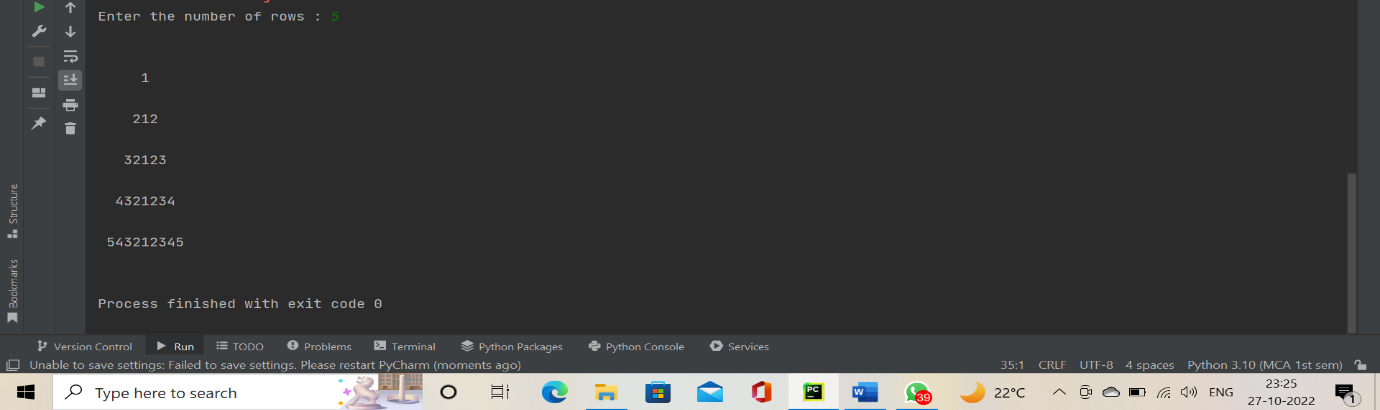
n=int(input("enter a number"))  
i = 1  
j = 0  
while (i <= n):  
 while (j <= i - 1):  
 print("\* ", end="")  
 j += 1  
 print("\r")  
 j = 0;  
 i += 1

**Output:**

 **Coding:**

rows = int(input("Enter the number of rows : "))  
print("\n")  
i = 1  
for i in range(i, rows+1):  
 j = i  
 for space in range(rows-i, -1, -1):  
 print(end=" ")  
 for j in range(j, 0, -1):  
 print(j,end="")  
 for j in range(j+1, i+1):  
 print(j,end="")  
 print("\n")

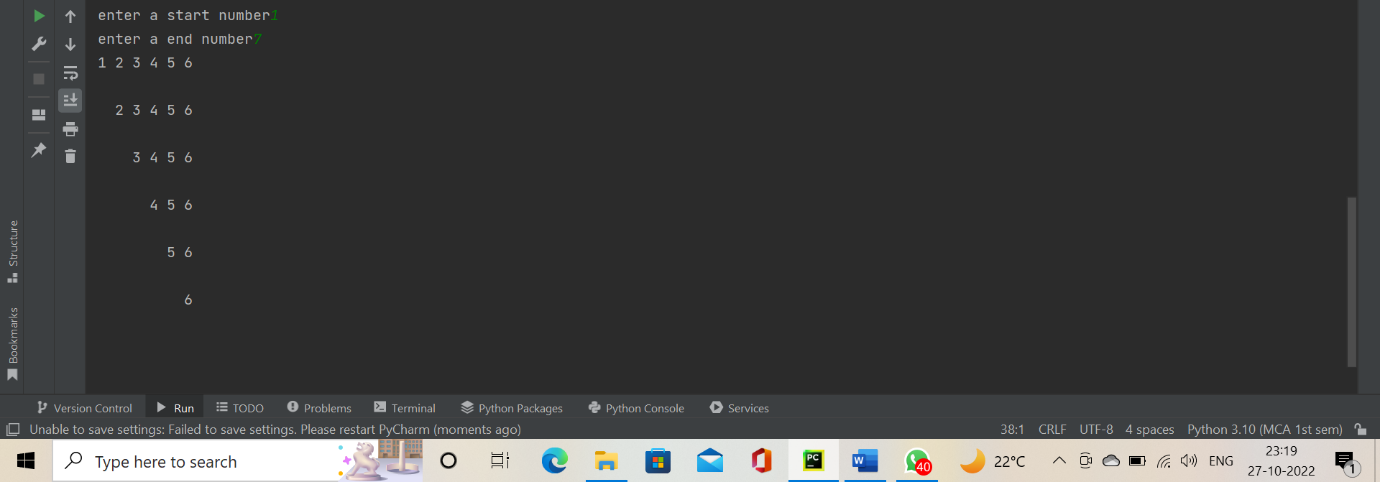
**Output:**



**Coding:**

j=int(input("enter a start number"))  
n=int(input("enter a end number"))  
while j<=n:  
 for i in range (j,n):  
 print(i,end=" ")  
 j=j+1  
 print("\n")  
 for k in range(1,j):  
 print(" ",end=" ")

**Output:**



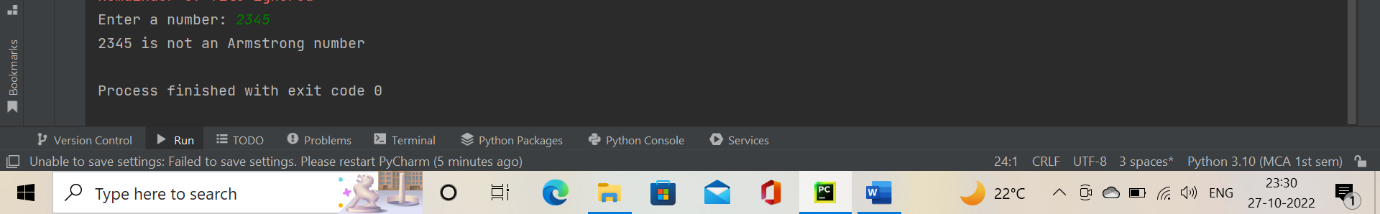
1. **Print Armstrong numbers in the range 1 to 1000. An Armstrong number is a number whose sum of the cubes of the digits is equal to the number itself.**

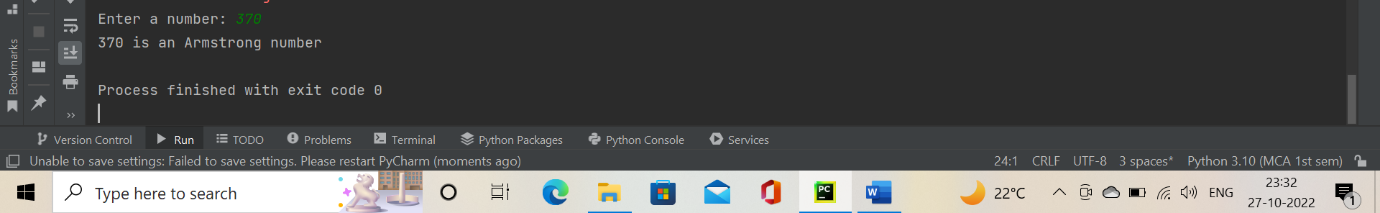
**For example, 370 = 33+73+03.**

**Coding:**

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

**Output:**



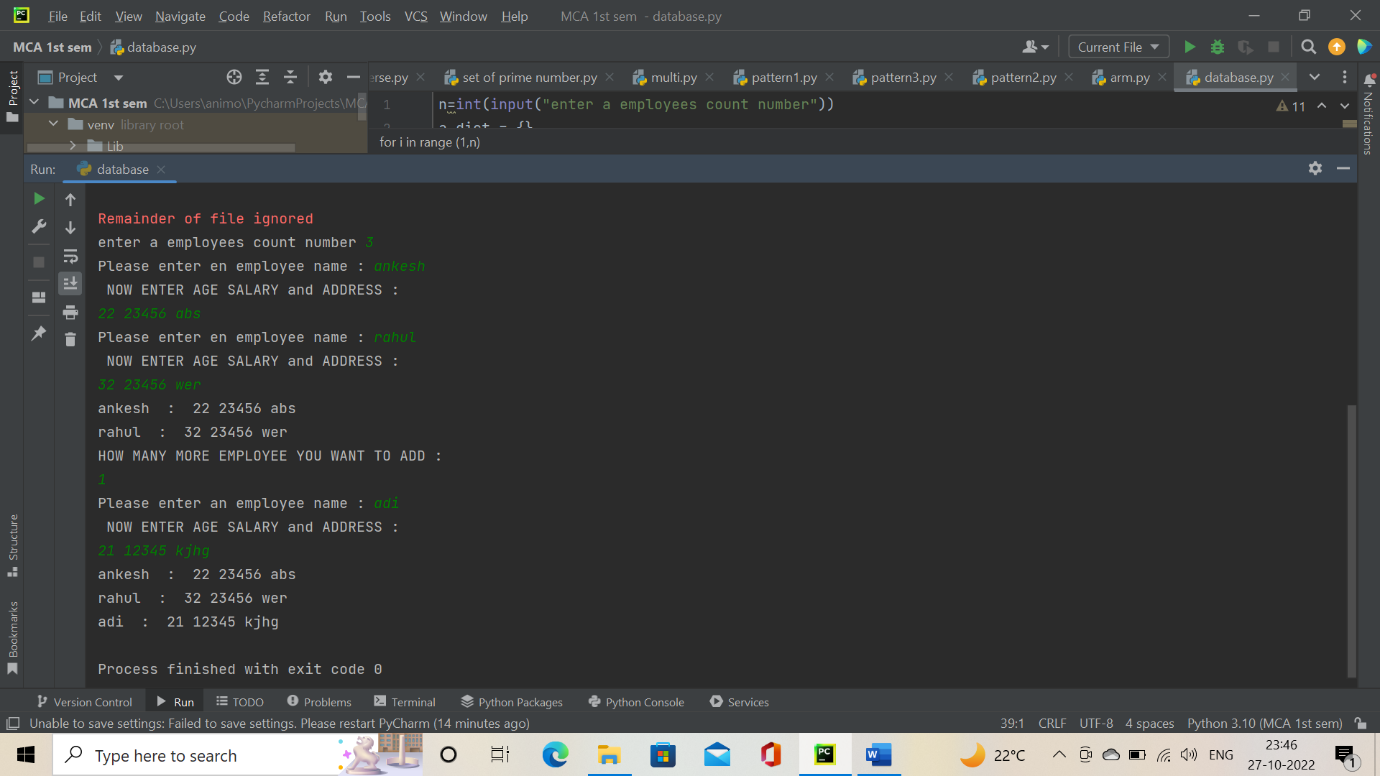


1. **Create an employee database having record for 5-10 employees in it. The attributes for an employee record are his/her name, age, salary, address. Take inputs from the user.**
2. **Print each record on a separate line**
3. **Update a record with given name.**

**Coding:**

n=int(input("enter a employees count number"))  
a\_dict = {}  
for i in range (1,n):  
 name=input("Please enter en employee name : ")  
 print(" NOW ENTER AGE SALARY and ADDRESS : ")  
 a\_dict[name] = input()  
for key, value in a\_dict.items():  
 print(key, ' : ', value)  
print("HOW MANY MORE EMPLOYEE YOU WANT TO ADD :")  
choice=int(input())  
if(choice==1):  
 for i in range(0,choice):  
 name=input("Please enter an employee name : ")  
 print(" NOW ENTER AGE SALARY and ADDRESS : ")  
 a\_dict[name] = input()  
for key, value in a\_dict.items():  
 print(key, ' : ', value)

**Output:**



**Assignment: 4**

**Experiments**

1. **Use following predefined functions and interpret the results:**

**Min () and max ()**

**Bin (), Oct (), Hex ()**

**Pow ()**

**Eval () and exec ()**

**Chr () and Ord ()**

**Round ()**

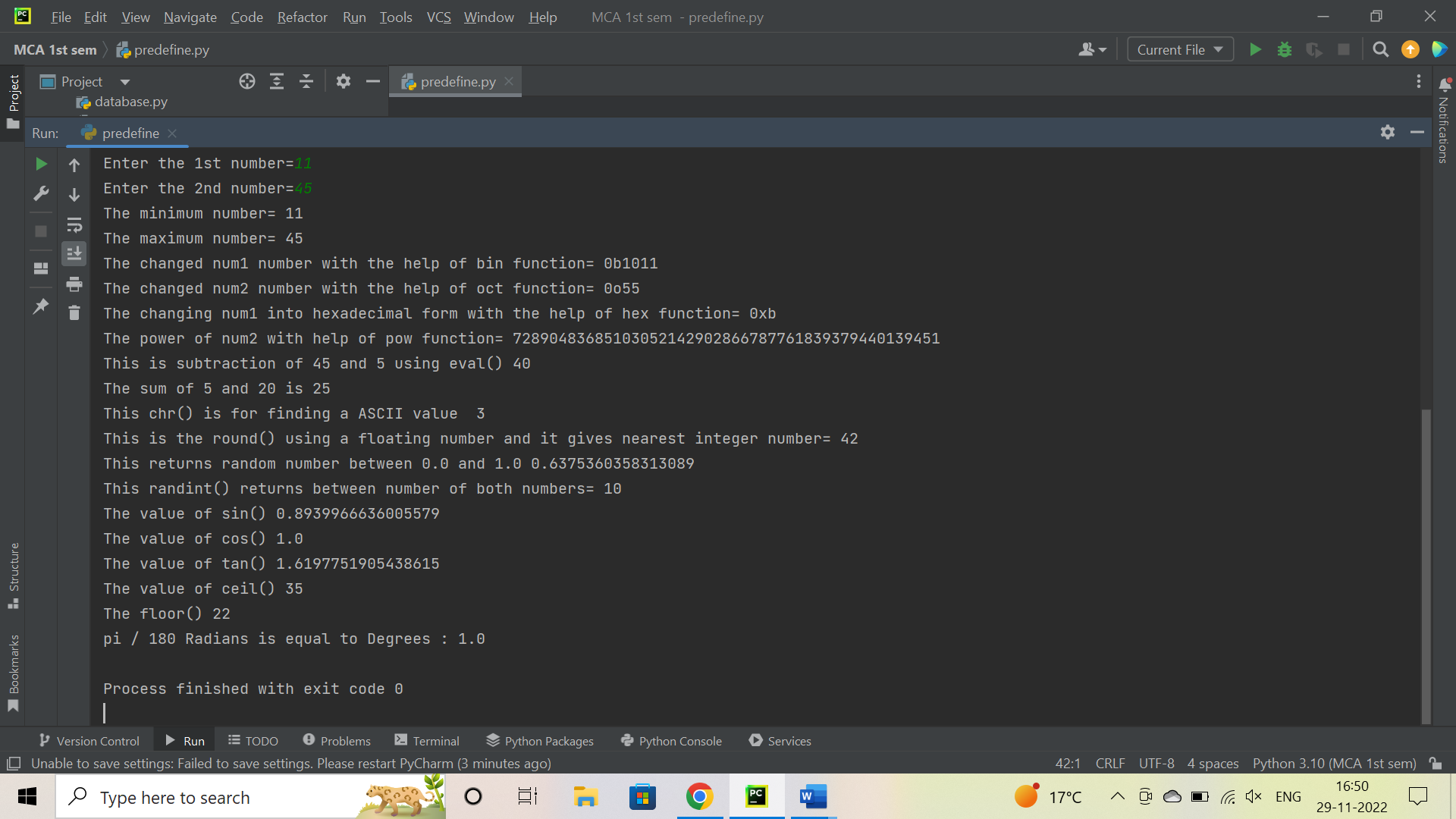
**Random (), rand int (), Import random module Random.rondom () Import math module**

**Sin (), Cos (), Tan (), Ceil (), Floor (), Degrees ().**

**Coding:**

num1=int(input("Enter the 1st number="))  
num2=int(input("Enter the 2nd number="))  
minimum=min(num1,num2)  
print("The minimum number=",minimum)  
maximum=max(num1,num2)  
print("The maximum number=",maximum)  
x=bin(num1)  
print("The changed num1 number with the help of bin function=",x)  
y=oct(num2)  
print("The changed num2 number with the help of oct function=",y)  
z=hex(num1)  
print("The changing num1 into hexadecimal form with the help of hex function=",z)  
power=pow(num1,num2)  
print("The power of num2 with help of pow function=",power)  
S="print('This is subtraction of 45 and 5 using eval()',45-5)"  
eval(S)  
program = 'print("The sum of 5 and 20 is", (5+20))'  
exec(program)  
y1=chr(51)  
print("This chr() is for finding a ASCII value ",y1)  
y2='A'  
ord(y2)  
x1= round(41.5)  
print("This is the round() using a floating number and it gives nearest integer number=",x1)  
import random  
print("This returns random number between 0.0 and 1.0",random.random())  
print("This randint() returns between number of both numbers=",random.randint(1,20))  
import math as m  
print("The value of sin()",m.sin(90))  
print("The value of cos()",m.cos(0))  
print("The value of tan()",m.tan(45))  
print("The value of ceil()",m.ceil(34.7))  
print("The floor()",m.floor(22.31))  
print("pi / 180 Radians is equal to Degrees : ", end ="")  
print (m.degrees(m.pi / 180))

**Output:**

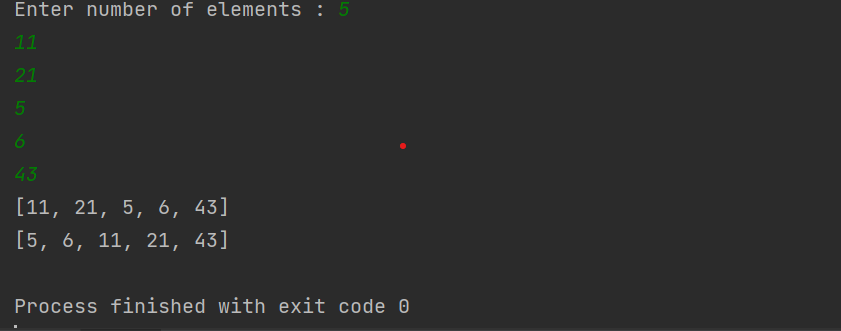


1. **Write a function to sort the contents of an integer list.**

**Coding:**

List = []  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
 ele = int(input())  
 List.append(ele)  
print(List)  
List.sort()  
print(List)

**Output:**

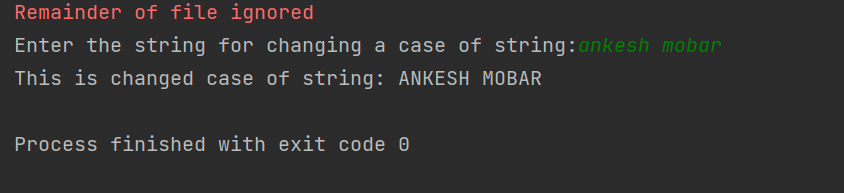


1. **Write a function to change the case of a given string.**

**Coding:**

Letter = input("Enter the string for changing a case of string:")  
L = Letter.upper()  
print("This is changed case of string:", L)

**Output:**

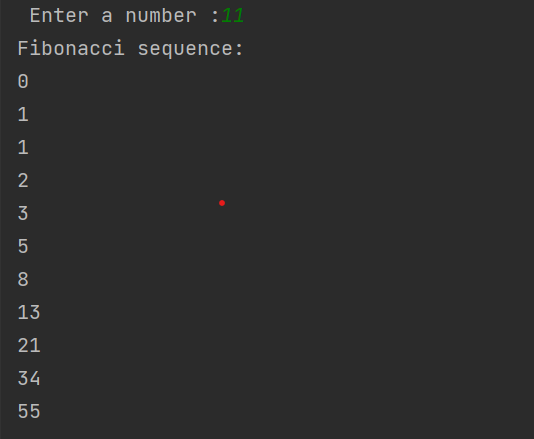


1. **The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Write a function recursive) to print n terms of this series based upon user input.**

**Coding:**

def recur\_fibo(n):  
 if n <= 1:  
 return n  
 else:  
 return(recur\_fibo(n-1) + recur\_fibo(n-2))  
Fiboseries = int(input(" Enter a number :"))  
if Fiboseries <= 0:  
 print(" enter a positive integer ")  
else:  
 print("Fibonacci sequence:")  
 for i in range(Fiboseries):  
 print(recur\_fibo(i))

**Output:**



**Assignment: 5**

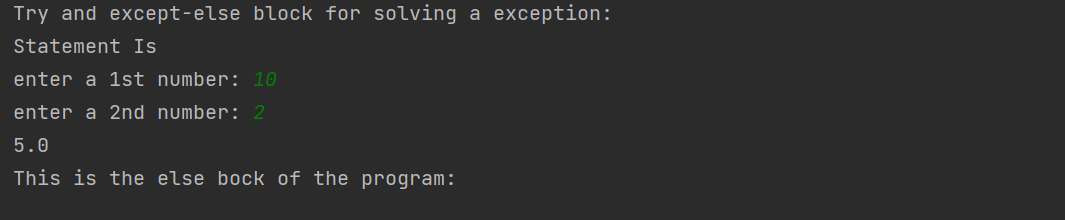
**Experiments**

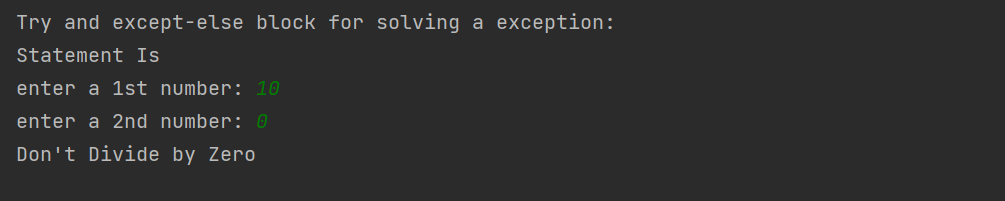
1. **Write a Python program to use try-except-else block.**

**Coding:**

print("Try and except-else block for solving a exception:")  
try:  
 print("Statement Is")  
 num1= int(input("enter a 1st number: "))  
 num2 = int(input("enter a 2nd number: "))  
 print(num1/num2)  
except ZeroDivisionError:  
 print("Don't Divide by Zero")  
except:  
 print("Please check your input number:")  
else:  
 print("This is the else bock of the program:")

**Output:**

****

****

1. **Write a Python program to use try-except-finally block.**

**Coding:**

try:  
 num1= int(input("enter a 1st number: "))  
 str= str(input("enter a 2nd number: "))  
except ZeroDivisionError:  
 print("Don't Divide by Zero")  
except NameError:  
 print("Don't Divide by Name")  
finally:  
 print("This is our final block!")

**Output:**

****

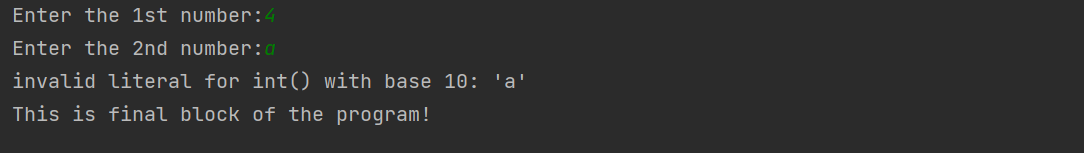
1. **Write a Python program to write your own exception and throw it.**

**Coding:**

try:  
 x=int(input("Enter the 1st number:"))  
 y=int(input("Enter the 2nd number:"))  
 print(x/y)  
except (ZeroDivisionError,ValueError) as msg:  
 print(msg)  
finally:  
 print("This is final block of the program!")

**Output:**

****

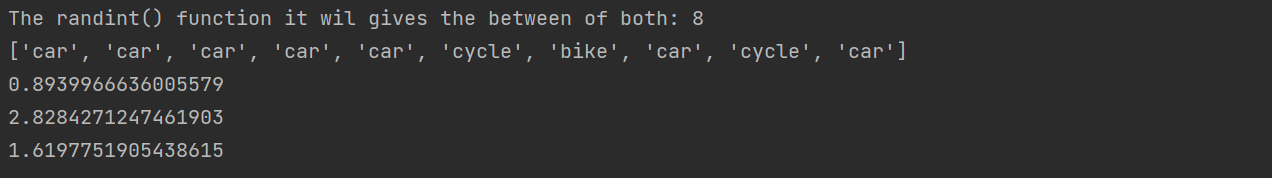
****

1. **Write a Python program to demonstrate the use of built-in modules random and math.**

**Coding:**

import random as r  
print("The randint() function it wil gives the between of both:",r.randint(1,8))  
li=["car","cycle","bike"]  
print(r.choices(li,weights=[5,1,1],k=10))  
import math as m  
print(m.sin(90))  
print(m.sqrt(8))  
print(m.tan(45))

**Output:**

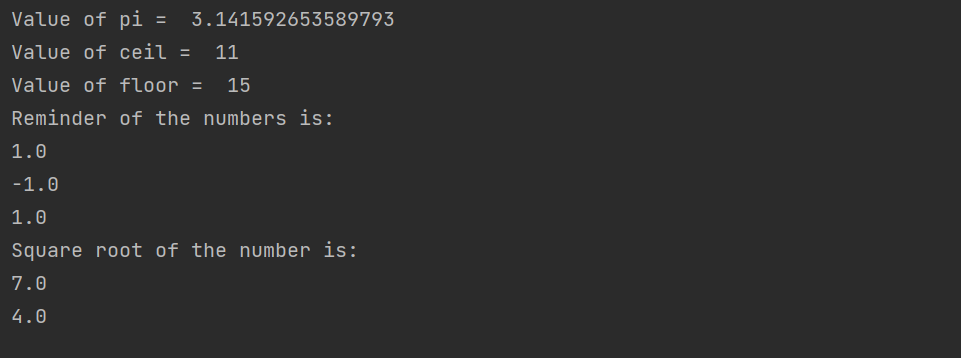
****

1. **Write a program to create a module and to use its functionality.**

**Coding:**

import math as m  
print("Value of pi = ", m.pi)  
a = m.ceil(11)  
b = m.floor(15)  
print("Value of ceil = ",a)  
print("Value of floor = ",b)  
print("Reminder of the numbers is: ")  
print (m.remainder(1, 10))  
print (m.remainder(15, 4))  
print (m.remainder(21, 5))  
print("Square root of the number is:")  
print (m.sqrt(49))  
print(m.sqrt(16))

**Output:**

****

**Assignment: 6**

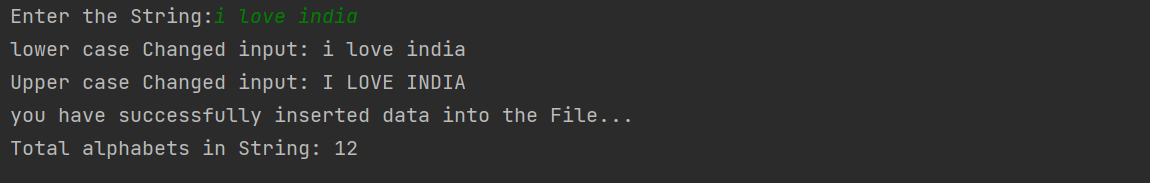
**Experiments**

1. **Write a program to count the number of English alphabets (lowercase and uppercase) in a text file.**

**Coding:**

s = input("Enter the String:")  
f = open("alphabets.txt","w")  
D = f.write(s)  
L = s.lower()  
U = s.upper()  
print("lower case Changed input:",L)  
print("Upper case Changed input:",U)  
print("you have successfully inserted data into the File...")  
count=0  
for i in s:  
 count=count+1  
print("Total alphabets in String:",count)

**Output:**

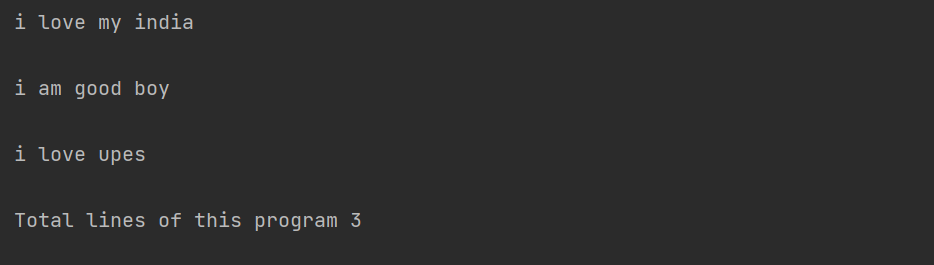
****

1. **Write a program to count the number of lines in a text file.**

**Coding:**

f=open("my.txt","r+")  
l=f.readlines()  
count=0  
for i in l:  
 print(i)  
 count=count+1  
print("Total lines of this program",count)  
f.close()

**Output:**

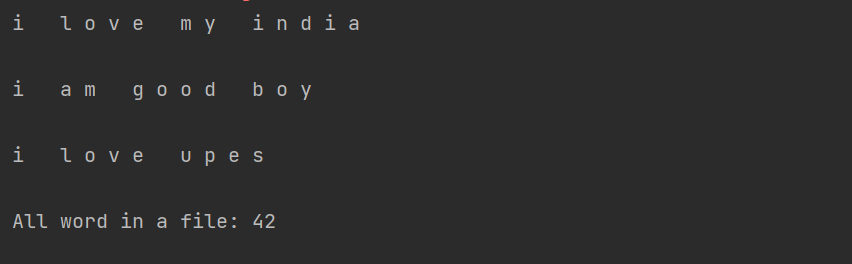
****

1. **Write a program to count the occurrences of a specific word in a text file.**

**Coding:**

f=open("my.txt","r+")  
l=f.readlines()  
count=0  
for i in l:  
 for word in i:  
 print(word,end=" ")  
 count=count+1  
 print()  
print("All word in a file:",count)  
f.close()

**Output:**

****

1. **Write a Python program to take input from the user and record it in a file. Once the input is over, program also shows the content of a file.**

**Coding:**

data=input("The record in a string:")  
f=open("my.txt",'w')  
f.write(data)  
print(data)

**Output:**

****

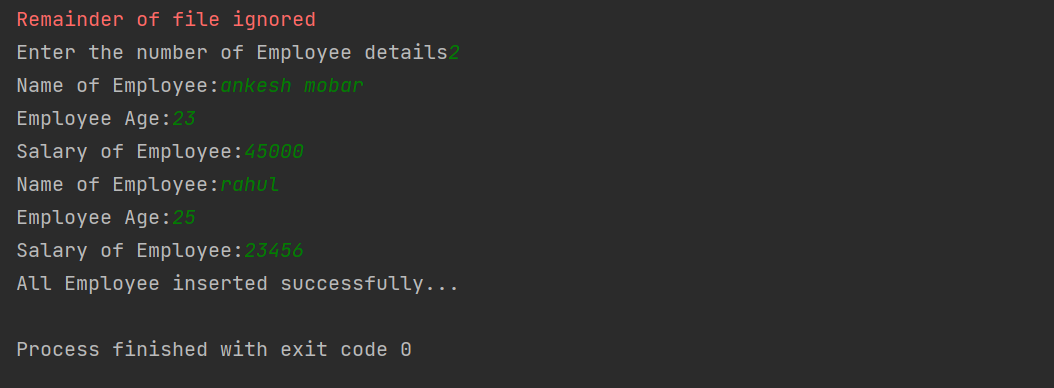
1. **Create a small database of employees that contains following attributes: Name, age, salary**

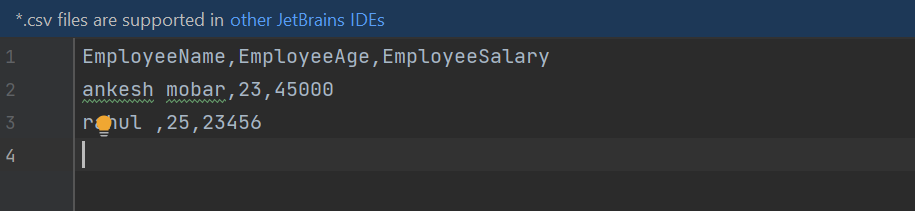
**Write a menu-driven program to**

1. **(a) List all the employees**
2. **(b) appending a new employee**
3. **(c) Searching for a specific employee**

**Coding:**

import csv  
with open("back.csv",'w',newline='') as f:  
 w=csv.writer(f)  
 w.writerow(["EmployeeName","EmployeeAge","EmployeeSalary"])  
 num=int(input("Enter the number of Employee details"))  
 for i in range(num):  
 EmployeeName=input("Name of Employee:")  
 EmployeeAge=int(input("Employee Age:"))  
 EmployeeSalary=int(input("Salary of Employee:"))  
 w.writerow([EmployeeName , EmployeeAge , EmployeeSalary])  
 print("All Employee inserted successfully...")

1. **Output:**
2. ****

****

**Assignment: 7**

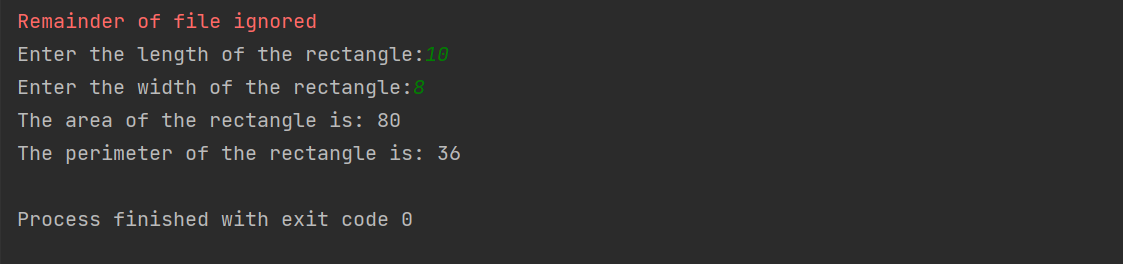
**Experiments**

1. **Write a class named Rectangle. Take the appropriate attributes and methods. Instantiate this class to use its attributes.**

**Coding:**

class Rectangle:  
 def \_\_init\_\_(self, length, width):  
 self.length = length  
 self.width = width  
  
 def area(self):  
 return self.length \* self.width  
  
 def perimeter(self):  
 return 2 \* (self.length + self.width)  
  
# Instantiate the Rectangle class  
length = int(input("Enter the length of the rectangle:"))  
width = int(input("Enter the width of the rectangle:"))  
  
rectangle = Rectangle(length, width)  
  
# Calculate the area and perimeter  
print("The area of the rectangle is:", rectangle.area())  
print("The perimeter of the rectangle is:", rectangle.perimeter())

**Output:**

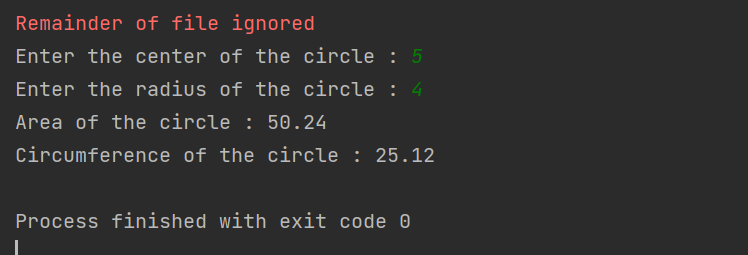


1. **Create a class Circle with attributes centre and radius. Add methods for area and circumference calculation. Now create circle objects to use these methods. Give the provision to take circle attributes from the user.**

**Coding:**

class Circle:  
 def \_\_init\_\_(self, center, radius):  
 self.center = center  
 self.radius = radius  
  
 def area(self):  
 return 3.14 \* self.radius \* self.radius  
  
 def circumference(self):  
 return 2 \* 3.14 \* self.radius  
  
  
c1 = Circle(input("Enter the center of the circle : "), int(input("Enter the radius of the circle : ")))  
print("Area of the circle :", c1.area())  
print("Circumference of the circle :", c1.circumference())

**Output:**



1. **Create a class named Account. An Account object can have following attributes:**

**Account Number**

**Account Holder**

**Account Type**

**Balance**

**credit\_account()**

**debit\_account()**

**get\_interest() (SI)**

**get\_account()**

**set\_account()**

**Add some class attributes to this Account class. Provide the appropriate constructor.**

**Create some sample bank account, use the listed methods and finally delete them.**

**Dynamically create account objects by navigating through a menu, such as:**

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

Create account

Withdraw amount

Deposit amount

Show Interest

Delete account

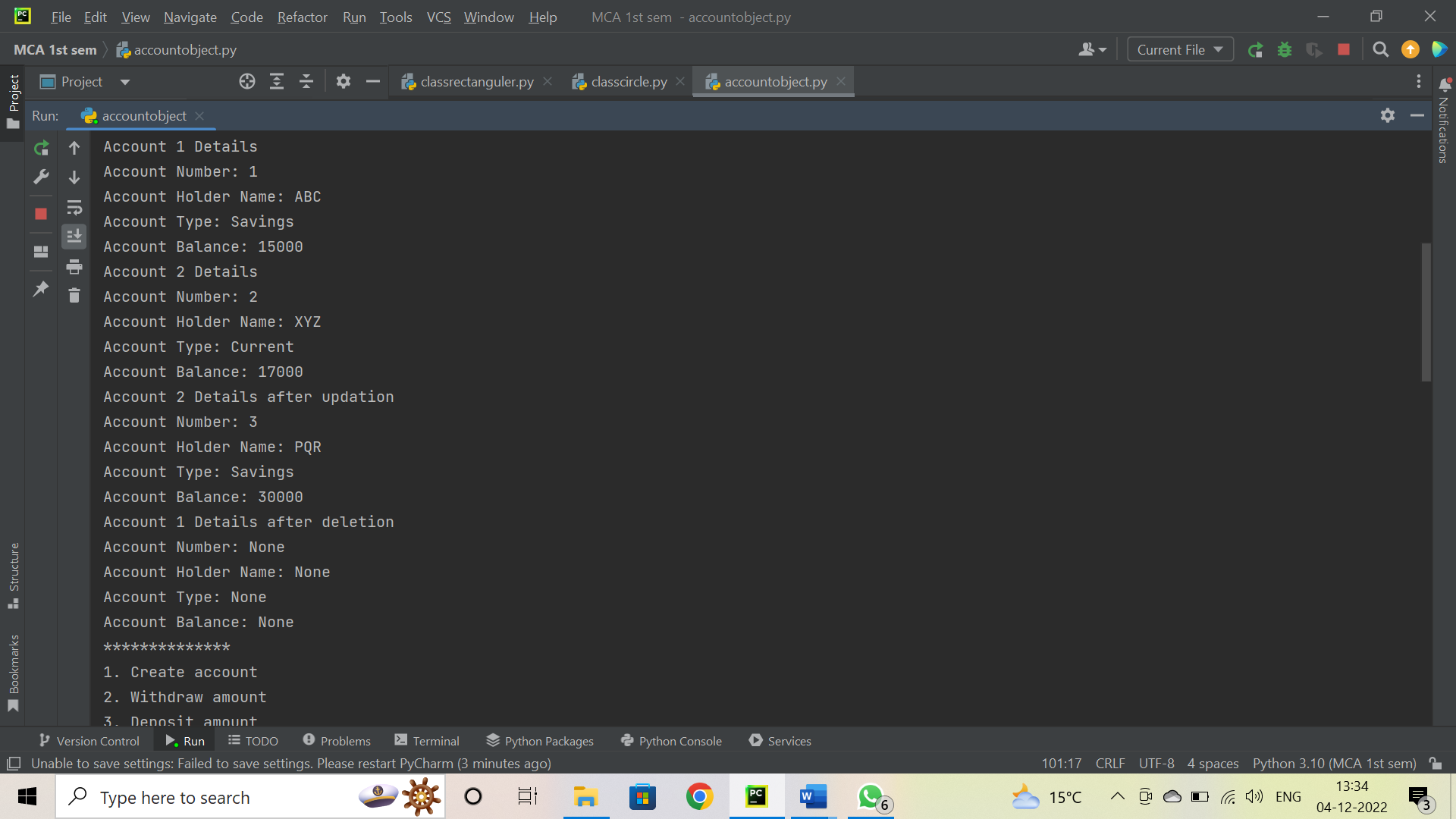
\*\*\*\*\*\*\*\*\*\*\*\*\*\*

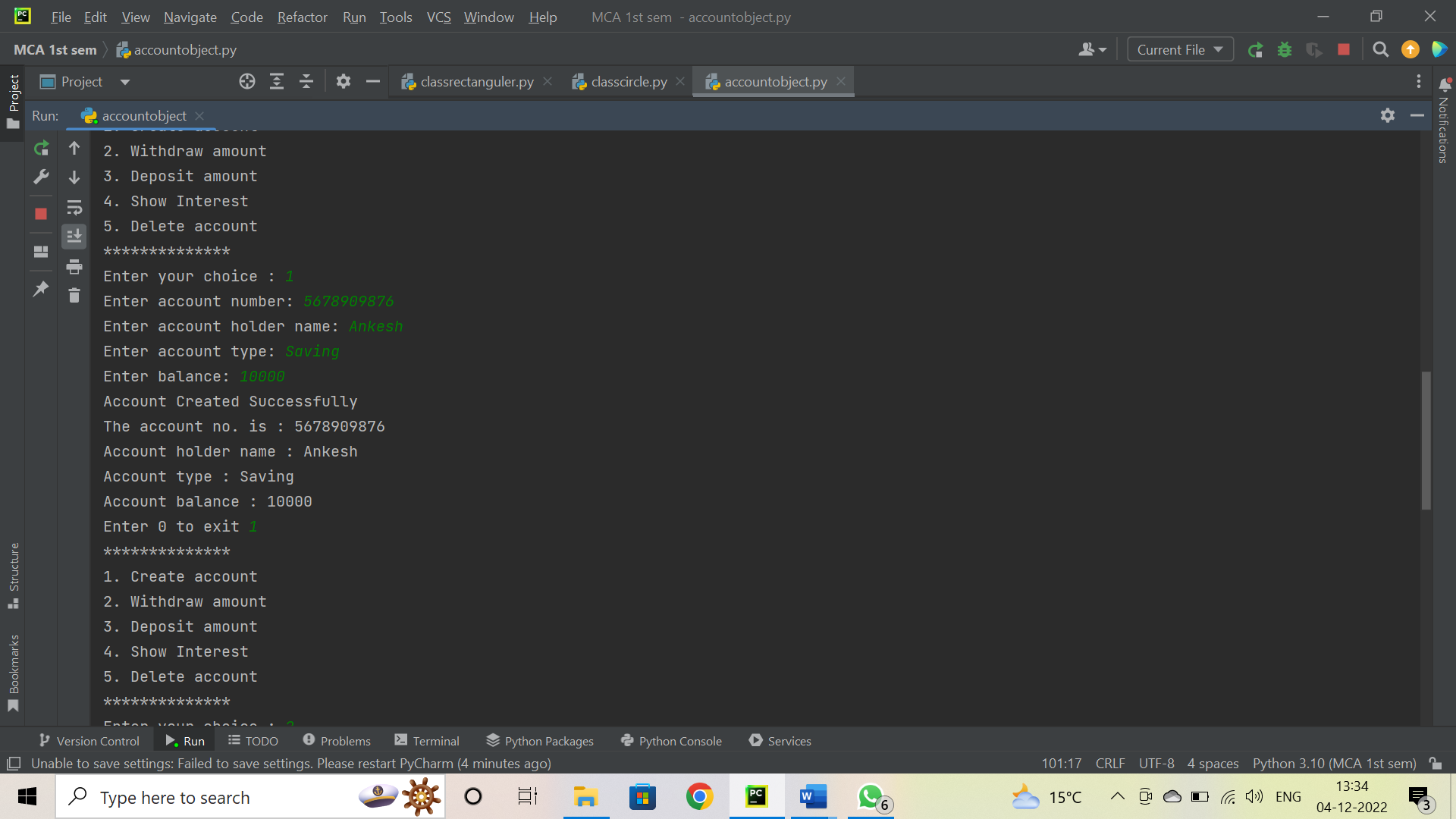
Enter your choice?

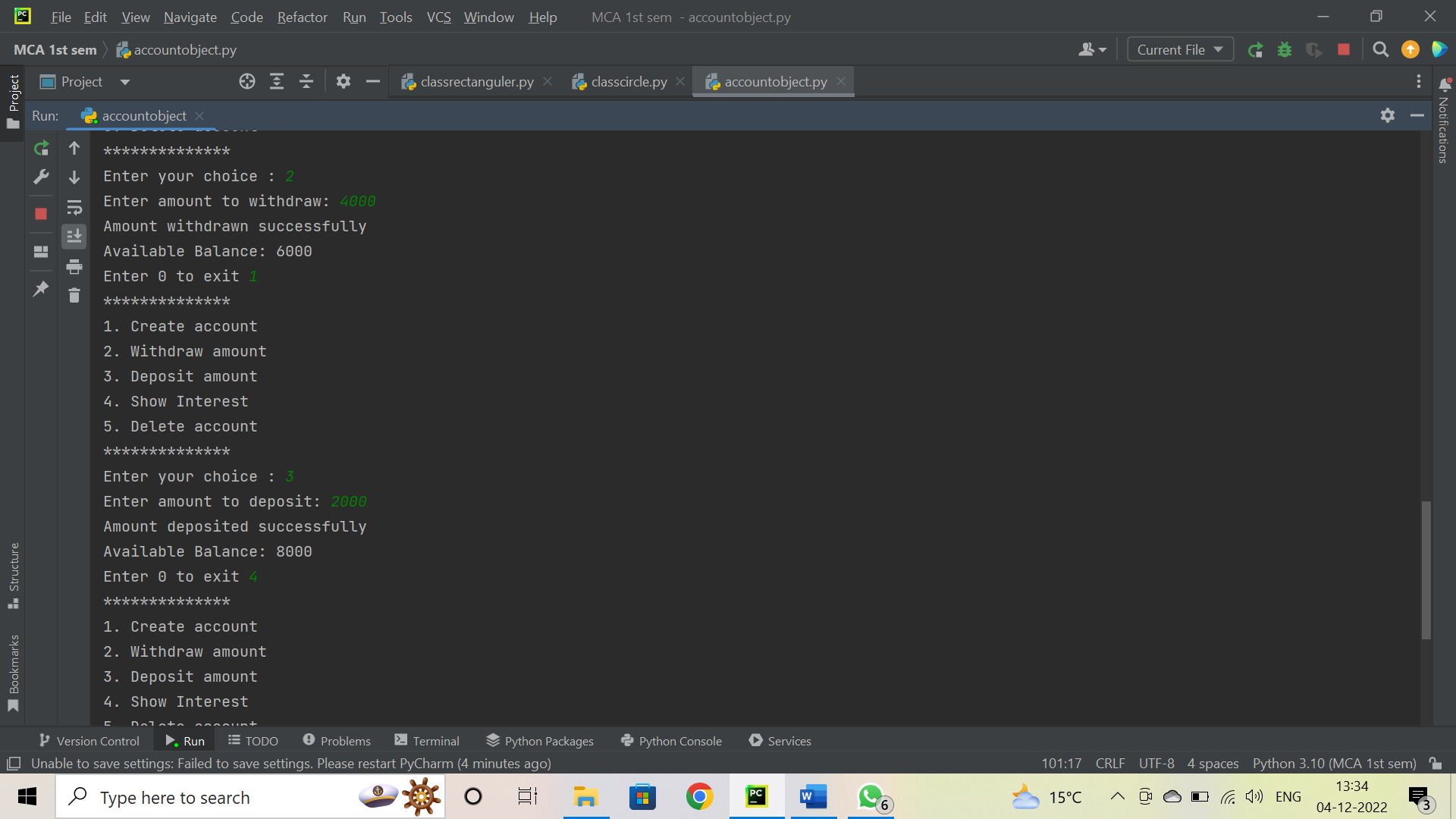
**Coding:**

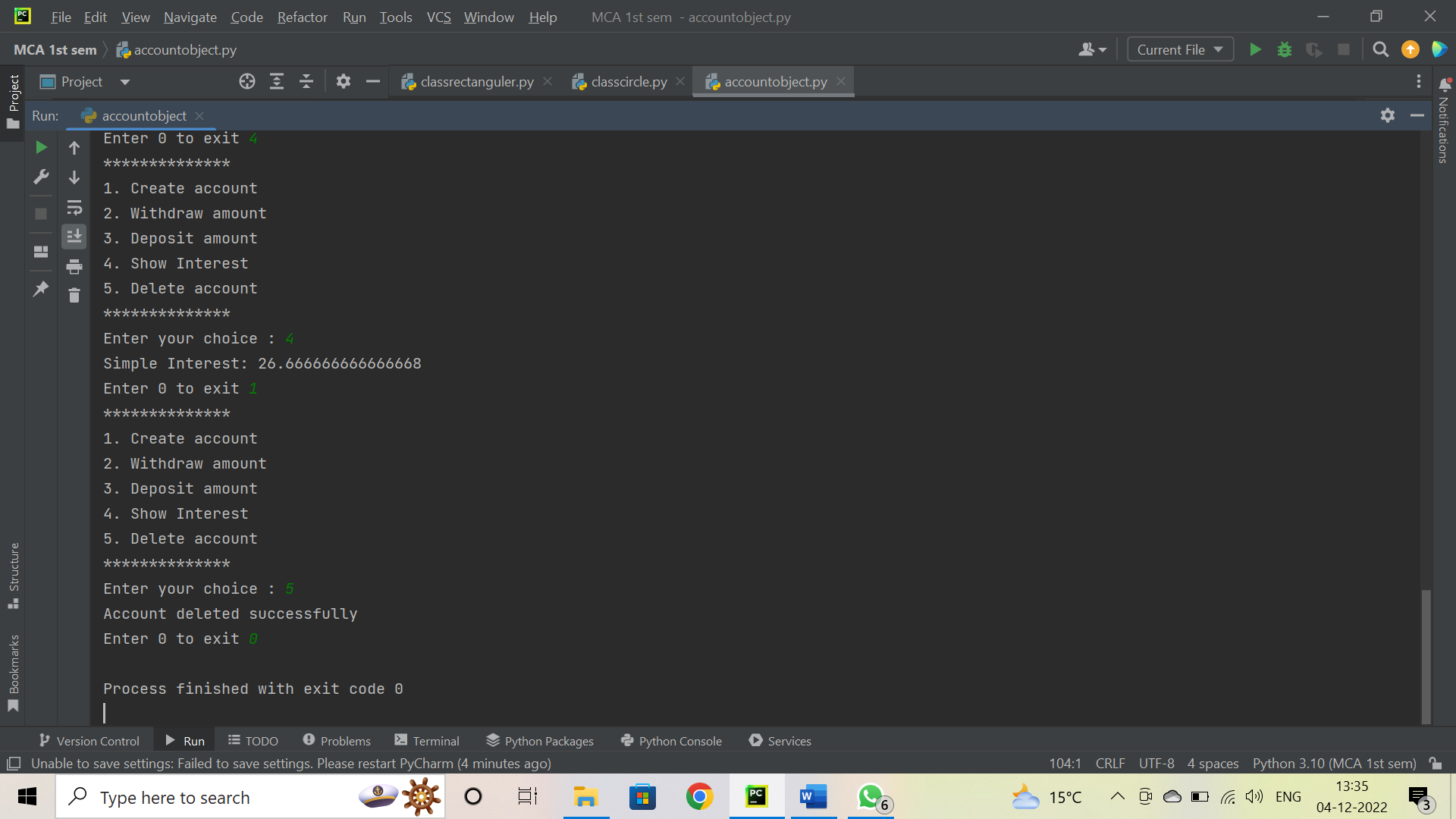
class Account:  
 # class attributes  
 bank\_name = "ABC Bank"  
 min\_deposit = 5000  
  
 # constructor  
 def \_\_init\_\_(self, acc\_num, acc\_holder, acc\_type, balance):  
 self.acc\_num = acc\_num  
 self.acc\_holder = acc\_holder  
 self.acc\_type = acc\_type  
 self.balance = balance  
  
 # Method to credit amount  
  
 def credit\_account(self, amount):  
 self.balance = self.balance + amount  
  
 # Method to debit amount  
  
 def debit\_account(self, amount):  
 self.balance = self.balance - amount  
  
 # Method to get interest  
  
 def get\_interest(self):  
 # SI = PTR  
 si = (self.balance \* 0.04 \* 1) / 12  
 return si  
  
 # Method to get account detail  
  
 def get\_account(self):  
 print("Account Number:", self.acc\_num)  
 print("Account Holder Name:", self.acc\_holder)  
 print("Account Type:", self.acc\_type)  
 print("Account Balance:", self.balance)  
  
 # Method to set account detail  
  
 def set\_account(self, acc\_num, acc\_holder, acc\_type, balance):  
 self.acc\_num = acc\_num  
 self.acc\_holder = acc\_holder  
 self.acc\_type = acc\_type  
 self.balance = balance  
  
 # Method to delete account  
  
 def delete\_account(self):  
 self.acc\_num = None  
 self.acc\_holder = None  
 self.acc\_type = None  
 self.balance = None  
  
 # creating a sample bank account  
  
  
acc1 = Account(1, "ABC", "Savings", 10000)  
acc2 = Account(2, "XYZ", "Current", 20000)  
  
# using the methods present in class  
acc1.credit\_account(5000)  
acc2.debit\_account(3000)  
  
# printing the account details  
print("Account 1 Details")  
acc1.get\_account()  
  
print("Account 2 Details")  
acc2.get\_account()  
  
# using set\_account method  
acc2.set\_account(3, "PQR", "Savings", 30000)  
  
print("Account 2 Details after updation")  
acc2.get\_account()  
  
# deleting account  
acc1.delete\_account()  
  
# printing the account details  
print("Account 1 Details after deletion")  
acc1.get\_account()  
  
  
# menu function  
def menu():  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("1. Create account")  
 print("2. Withdraw amount")  
 print("3. Deposit amount")  
 print("4. Show Interest")  
 print("5. Delete account")  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 choice = int(input("Enter your choice : "))  
 return choice  
  
  
# main function  
while True:  
 choice = menu()  
  
 # create account  
 if choice == 1:  
 acc\_num = int(input("Enter account number: "))  
 acc\_holder = input("Enter account holder name: ")  
 acc\_type = input("Enter account type: ")  
 balance = int(input("Enter balance: "))  
  
 # creating object of class  
 acc = Account(acc\_num, acc\_holder, acc\_type, balance)  
 print("Account Created Successfully")  
 print('The account no. is :',acc\_num)  
 print('Account holder name :',acc\_holder)  
 print('Account type :',acc\_type)  
 print('Account balance :',balance)  
  
 # withdraw amount  
 elif choice == 2:  
 amount = int(input("Enter amount to withdraw: "))  
 acc.debit\_account(amount)  
 print("Amount withdrawn successfully")  
 print("Available Balance:", acc.balance)  
  
 # deposit amount  
 elif choice == 3:  
 amount = int(input("Enter amount to deposit: "))  
 acc.credit\_account(amount)  
 print("Amount deposited successfully")  
 print("Available Balance:", acc.balance)  
  
 # show interest  
 elif choice == 4:  
 si = acc.get\_interest()  
 print("Simple Interest:", si)  
  
 # delete account  
 elif choice == 5:  
 acc.delete\_account()  
 print("Account deleted successfully")  
  
  
 # invalid choice  
 else:  
 print("Invalid Choice")  
  
 # exit condition  
 exit\_choice = int(input("Enter 0 to exit "))  
 if exit\_choice == 0:  
 break

**Output:**









**Assignment: 8**

**Experiments**

1. **Write a class named Rectangle. Take length and breadth as attributes and area as methods. Create a subclass Square with the only attribute as side and override the area method. Create Rectangle and Square object statically and dynamically and make use of area method.**

**Coding:**

class Rectangle:  
 # Constructor  
 def \_\_init\_\_(self, length, breadth):  
 self.length = length  
 self.breadth = breadth  
  
 # Area of Rectangle  
  
 def area(self):  
 return self.length \* self.breadth  
  
 # SubClass  
  
  
class Square(Rectangle):  
 # Constructor  
 def \_\_init\_\_(self, side):  
 self.length = side  
 self.breadth = side  
  
 # Objects of Rectangle  
  
  
a = int(input("Enter the length of Rectangle: "))  
b = int(input("Enter the breadth of Rectangle: "))  
obj1 = Rectangle(a, b)  
  
# Area of Rectangle  
print("Area of Rectangle:", obj1.area())  
  
# Object of Square  
c = int(input("Enter the side of Square: "))  
obj2 = Square(c)  
  
# Area of Square  
print("Area of Square:", obj2.area())

**Output:**

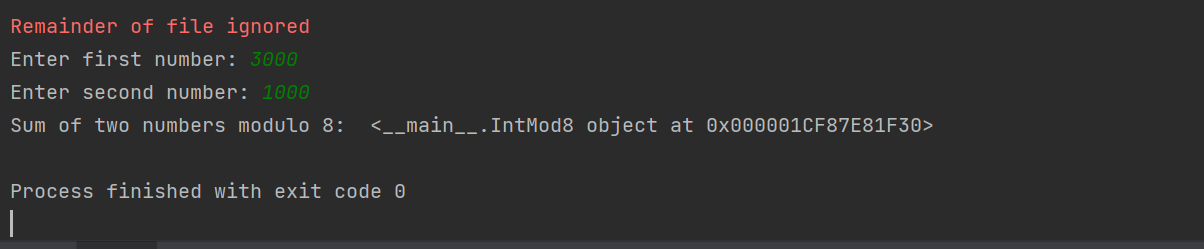


1. **Overload the + (addition) operator to restrict the addition of two integers to modulo 8.**

**Coding:**

class IntMod8:  
 def \_\_init\_\_(self,value):  
 self.value=value  
  
 def \_\_add\_\_(self,other):  
 return IntMod8(self.value%8 + other.value%8)  
  
x = IntMod8(int(input("Enter first number: ")))  
y = IntMod8(int(input("Enter second number: ")))  
  
print("Sum of two numbers modulo 8: ",x+y)

**Output:**



1. **Implement a Stack class using a list type. Provide push() and pop() operations. Demonstrate the usage.**

**Coding:**

class Stack:  
 def \_\_init\_\_(self):  
 self.stack = []  
  
 def push(self, val):  
 self.stack.append(val)  
  
 def pop(self):  
 return self.stack.pop()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 s = Stack()  
 # Taking input from user  
 n = int(input("Enter the number of elements:"))  
 print("Enter the elements:")  
 for i in range(n):  
 s.push(input())  
  
 print("Stack elements are:")  
 while s.stack:  
 print(s.pop())

**Output:**



**Assignment: 9**

**Experiments**

1. **Write your own iterator class and iterate through it (without generator function).**

**Coding:**

class MyIterator:  
 def \_\_init\_\_(self, list\_of\_numbers):  
 self.list\_of\_numbers = list\_of\_numbers  
 self.index = 0  
  
 def \_\_iter\_\_(self):  
 return self  
  
 def \_\_next\_\_(self):  
 if self.index == len(self.list\_of\_numbers):  
 raise StopIteration  
 else:  
 index = self.index  
 self.index += 1  
 return self.list\_of\_numbers[index]  
  
#Taking input from user  
list\_of\_numbers = [int(x) for x in input('Enter the numbers : ').split()]  
# Creating an instance of MyIterator  
my\_iterator = MyIterator(list\_of\_numbers)  
#Iterating through the iterator  
for num in my\_iterator:  
 print(num)

**Output:**



1. **Write your iterator by using generator function.**

**Coding:**

def my\_generator(n):  
 for i in range(n):  
 yield i  
  
n = int(input("Enter a number: "))  
  
for item in my\_generator(n):  
 print(item)

**Output:**



1. **Write a function that calculates simple interest. Now decorate the output of this function by returning the modified amount.**

**Coding:**

def calculate\_simple\_interest(principal, rate, time):  
 si = (principal \* rate \* time) / 100  
 return si  
  
def decorate\_amount(amount):  
 return 'Amount : ' + str(amount)  
  
principal = float(input('Enter principal amount : '))  
rate = float(input('Enter rate of interest : '))  
time = float(input('Enter time in years : '))  
  
simple\_interest = calculate\_simple\_interest(principal, rate, time)  
print(decorate\_amount(simple\_interest))

**Output:**

