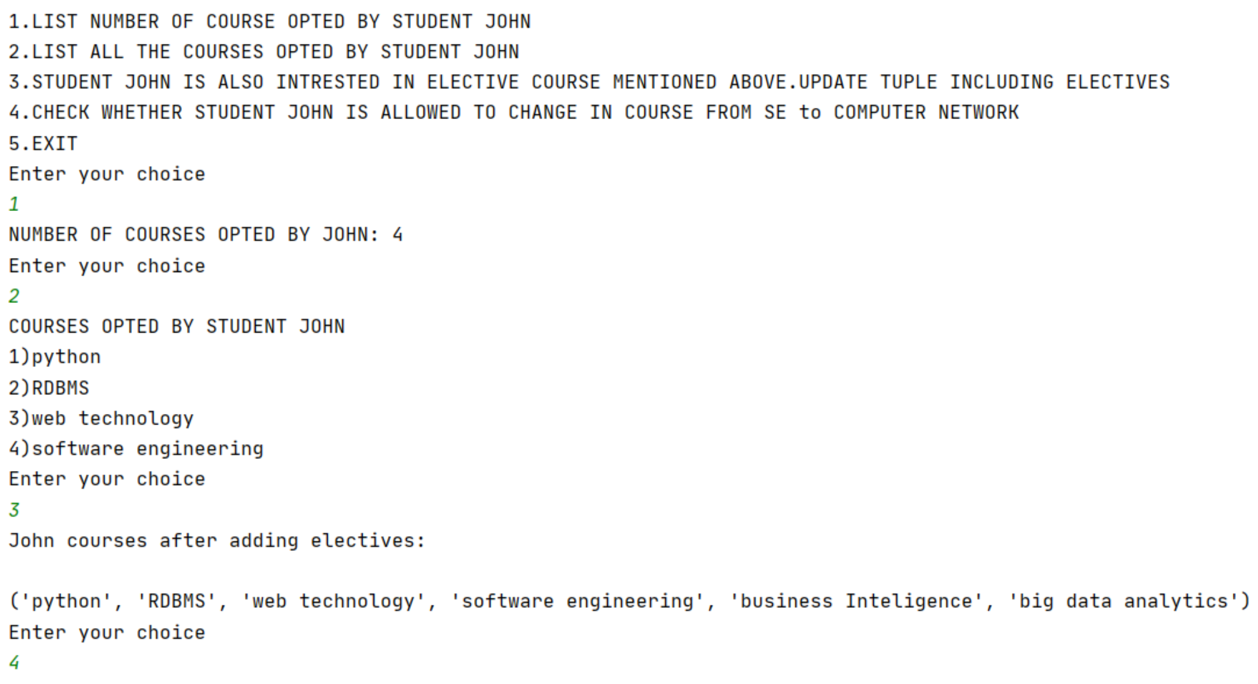
```
"""---------------------------------------------------------------------------------
Lab13.py
13.Write a python program using tuple to satisfy following business requirements:
   a) List the number of courses opted by student “John”
   b) List all the courses opted by student “John”
   c) Student “John” is also interested in elective course mentioned above. Print the updated
tuple including electives.
   d) Check whether student “john” is allowed to change his course from SE to Computer
network. Consider the list of courses opted by a student “john” and available electives as a
part of student Management System.
   Courses: (“Python Programming”, ”RDBMS”, ”Web Technology”,
   “Software Engineering”)
   Electives:(”Business Intelligence”, Big Data Analytics”)
Regno: 2117108
04/04/2022
---------------------------------------------------------------------------------"""

courses = ("python", "RDBMS", "web technology", "software engineering")
electives = ("business Inteligence", "big data analytics")

print("1.LIST NUMBER OF COURSE OPTED BY STUDENT JOHN")
print("2.LIST ALL THE COURSES OPTED BY STUDENT JOHN")
print("3.STUDENT JOHN IS ALSO INTRESTED IN ELECTIVE COURSE MENTIONED
ABOVE.UPDATE TUPLE INCLUDING ELECTIVES")
print("4.CHECK WHETHER STUDENT JOHN IS ALLOWED TO CHANGE IN COURSE
FROM SE to COMPUTER NETWORK")
print("5.EXIT")
while True:
   ch = int(input("Enter your choice\n"))
   if ch == 1:
      print("NUMBER OF COURSES OPTED BY JOHN:", len(courses))
   elif ch == 2:
      print("COURSES OPTED BY STUDENT JOHN")
      i = 0
      for course in courses:
         i += 1
         print(f"{i}){course}")
   elif ch == 3:
      listcourse = list(courses)
      listele = list(electives)
      for ele in listele:
         listcourse.append(ele)
      course = tuple(listcourse)
      print("John courses after adding electives:\n")
      print(course)
   elif ch == 4:
      listcourse = list(course)
      for i in range(0, len(listcourse)):
         if listcourse[i] == 'software engineering':
            listcourse[i] = 'COMPUTER NETWORKS'
            courses = tuple(listcourse)
        print("John courses after chaging:\n")
        print(courses)
    elif ch == 5:
        break
    else:
        print("INVALID CHOICE!!!")
```

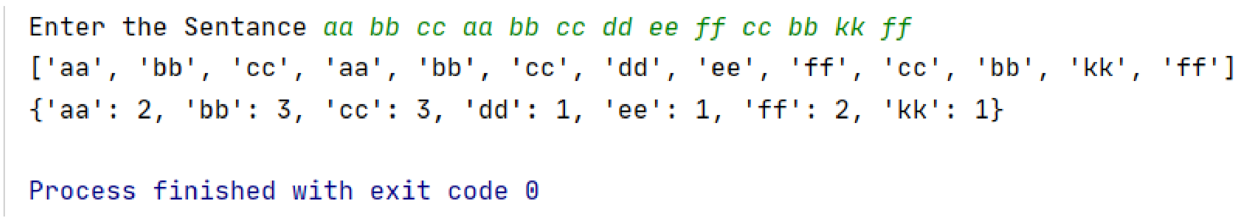


```
1.LIST NUMBER OF COURSE OPTED BY STUDENT JOHN
2.LIST ALL THE COURSES OPTED BY STUDENT JOHN
3.STUDENT JOHN IS ALSO INTRESTED IN ELECTIVE COURSE MENTIONED ABOVE.UPDATE TUPLE INCLUDING ELECTIVES
4.CHECK WHETHER STUDENT JOHN IS ALLOWED TO CHANGE IN COURSE FROM SE to COMPUTER NETWORK
5.EXIT
Enter your choice
1
NUMBER OF COURSES OPTED BY JOHN: 4
Enter your choice
2
COURSES OPTED BY STUDENT JOHN
1)python
2)RDBMS
3)web technology
4)software engineering
Enter your choice
3
John courses after adding electives:

('python', 'RDBMS', 'web technology', 'software engineering', 'business Inteligence', 'big data analytics')
Enter your choice
4
```

```python
'''-----------------------------------------------------------------------------
Lab14.py
14.Write a Python program to input 'n' names and phone numbers to store it a dictionary
and print the phone number of a particular name
Regno: 2117108
04/04/2022
----------------------------------------------------------------------------- '''

n = int(input("ENTER THE NO OF PEOPLE:"))
dict = {}
for i in range(n):
    keys = input("ENTER THE NAME:")
    def check():
        values = input("ENTER PHONE NUMBER:")
        if(len(values)!=10):
            print("NOT 10 DIGITS,\nEnter again")
            check()
        else:
            values=int(values)
            dict[keys]= values
    check()

for i in dict:
    print(dict)
    break

flag = 0
name = input("ENTER THE NAME TO FIND PHONE NUMBER:")
for key in dict:
    if name in key:
        print(f"PHONE NUMBER OF {name} = {dict[key]}")
        flag = 1
if flag == 0:
    print("NAME IS NOT FOUND")
```

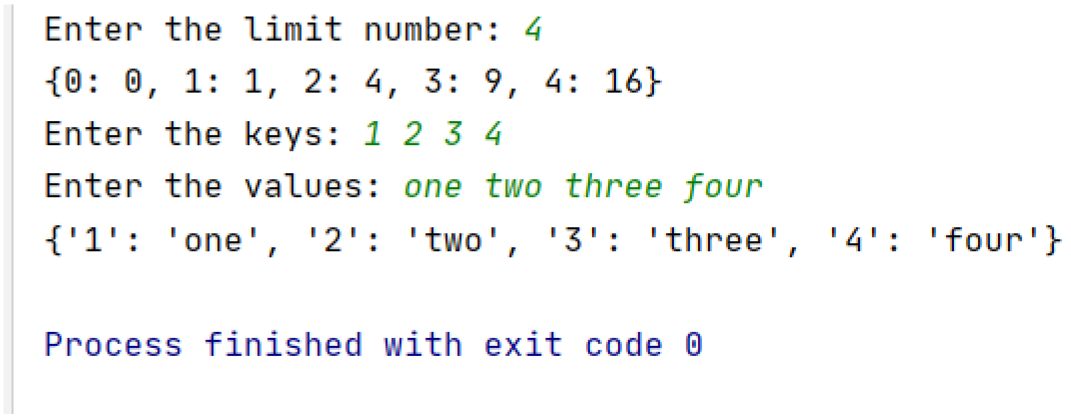
```
ENTER THE NO OF PEOPLE:2
ENTER THE NAME:Lailesh
ENTER PHONE NUMBER:998635
NOT 10 DIGITS,
Enter again
ENTER PHONE NUMBER:9961003680
ENTER THE NAME:Preetham
ENTER PHONE NUMBER:8673426632
{'Lailesh': 9961003680, 'Preetham': 8673426632}
ENTER THE NAME TO FIND PHONE NUMBER:Lailesh
PHONE NUMBER OF Lailesh = 9961003680

Process finished with exit code 0
```

```
"""-------------------------------------------------------------------------------
lab15.py
15.Write a function called string_dict that will take as parameter a string. The string can have
alphabets, spaces, question marks, periods and apostrophes only. The function returns a
dictionary. The keys of the dictionary should be the words from the original string, and the
values should be the frequency of that word.
Regno:2117108
05/04/2022
-------------------------------------------------------------------------------"""

def StringToList(Str1):
    list1 = Str1.split(' ')
    return list1

def ResultDictionary(li):
    f = {}
    for val in li:
        flag = True
        if not val.isalnum() and '?' not in val and '.' not in val:
            flag = False

        if flag:
            if val in f.keys():
                f[val] = f[val] + 1
            else:
                f[val] = 1
        else:
            print(val,"Not added To list")

    return f

Main_string = input("Enter the Sentance ")
list1 = []
list1 = StringToList(Main_string)
print(list1)
final_Result = {}
final_Result = ResultDictionary(list1)
print(final_Result)
```



```
Enter the Sentance aa bb cc aa bb cc dd ee ff cc bb kk ff
['aa', 'bb', 'cc', 'aa', 'bb', 'cc', 'dd', 'ee', 'ff', 'cc', 'bb', 'kk', 'ff']
{'aa': 2, 'bb': 3, 'cc': 3, 'dd': 1, 'ee': 1, 'ff': 2, 'kk': 1}

Process finished with exit code 0
```

```
"""---------------------------------------------------------------------------------
Lab16.py
16.Write a python script
1) To generate and print a dictionary that contains a number(between 1 and n) in the form
(x,x*x) .
2) To Map two list into dictionary.
Regno:2117108
06/04/2022
---------------------------------------------------------------------------------"""

final_dict = {}

n = int(input("Enter the limit number: "))

for i in range(n + 1):
    final_dict[i] = i * i

print(final_dict)

list1 = [item for item in input("Enter the keys: ").split()]
list2 = [item for item in input("Enter the values: ").split()]
final_dict_2 = dict(zip(list1, list2))

print(final_dict_2)
```

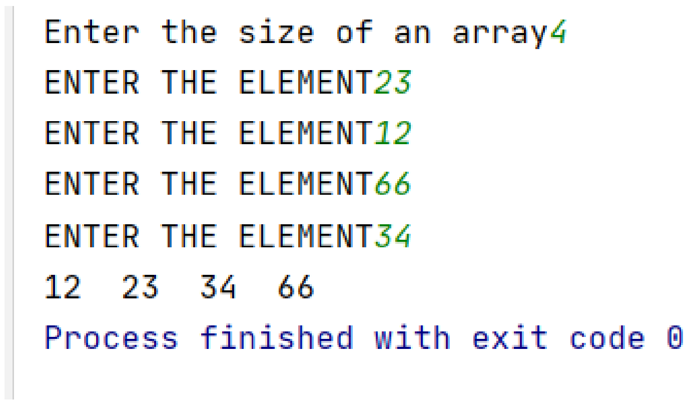


```
Enter the limit number: 4
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
Enter the keys: 1 2 3 4
Enter the values: one two three four
{'1': 'one', '2': 'two', '3': 'three', '4': 'four'}

Process finished with exit code 0
```

```
'''--------------------------------------------------------------------------------
Lab17.py
17.Write a Python program to sort the elements in the array using bubble sort technique and
display the elements in descending order
Regno: 2117108
6/04/2022
-------------------------------------------------------------------------------- '''
import module_sort
from array import array

n =int(input("Enter the size of an array"))
element = array('i')
for i in range(0,n):
    ele = int(input("ENTER THE ELEMENT"))
    element.append(ele)
module_sort.bubble_sort(element)
for i in range(len(element)):
    print(element[i],end="\t")
```

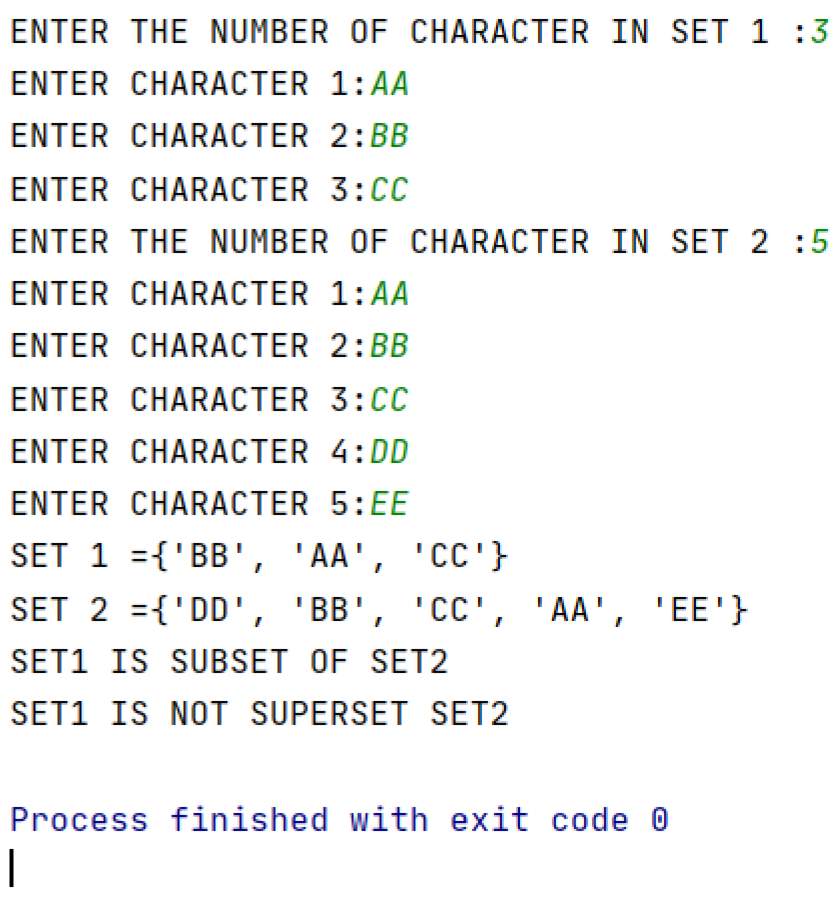
module_sort.py

```
def bubble_sort(b):
    for i in range (0,len(b)):
        for j in range(0,len(b)-i-1):
            if b[j] > b[j+1]:
                temp = b[j]
                b[j] = b[j+1]
                b[j+1] = temp
    return b
```



```
Enter the size of an array4
ENTER THE ELEMENT23
ENTER THE ELEMENT12
ENTER THE ELEMENT66
ENTER THE ELEMENT34
12  23  34  66
Process finished with exit code 0
```

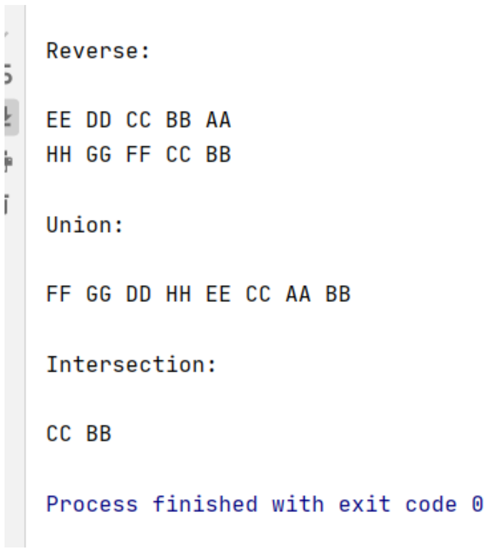
```
'''-------------------------------------------------------------------------------
Lab18.py
Write a python program to check whether the given is subset of a set or a super set of a set
Regno: 2117108
06/04/2022
------------------------------------------------------------------------------- '''

set1 = set()
set2 = set()

num_char = int(input("ENTER THE NUMBER OF CHARACTER IN SET 1 :"))
for i in range(num_char):
        ele = input(f"ENTER CHARACTER {i+1}:")
        set1.add(ele)

num_char = int(input("ENTER THE NUMBER OF CHARACTER IN SET 2 :"))
for i in range(num_char):
        ele = input(f"ENTER CHARACTER {i+1}:")
        set2.add(ele)

print(f"SET 1 ={set1}")
print(f"SET 2 ={set2}")

if set1.issubset(set2):
    print("SET1 IS SUBSET OF SET2")
else:
    print("SET1 IS NOT SUBSET OF SET2")

if set1.issuperset(set2):
    print("SET1 IS SUPERSET OF SET2")
else:
    print("SET1 IS NOT SUPERSET SET2")
```



```
ENTER THE NUMBER OF CHARACTER IN SET 1 :3
ENTER CHARACTER 1:AA
ENTER CHARACTER 2:BB
ENTER CHARACTER 3:CC
ENTER THE NUMBER OF CHARACTER IN SET 2 :5
ENTER CHARACTER 1:AA
ENTER CHARACTER 2:BB
ENTER CHARACTER 3:CC
ENTER CHARACTER 4:DD
ENTER CHARACTER 5:EE
SET 1 ={'BB', 'AA', 'CC'}
SET 2 ={'DD', 'BB', 'CC', 'AA', 'EE'}
SET1 IS SUBSET OF SET2
SET1 IS NOT SUPERSET SET2

Process finished with exit code 0
```

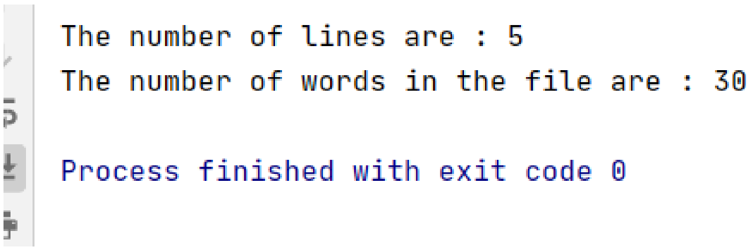
```
'''--------------------------------------------------------------------------------
Lab19.py
19.Write a python program to perform

i) Reverse in descending order, union in ascending order, intersection in ascending order
using the input present in the file.

ii) Print the output as well as save the file in the new file with file name as 'output program
<<programnumber>>_<<registernumber>> <<year>> <<month>> <<date>>.txt'

iii) Output of reverse, union, intersection should be printed in newline.

iv) Copy the program file from existing file destination to location where your input and out
file is present.

Regno:2117108
07/04/2022

--------------------------------------------------------------------------------'''
from datetime import datetime as dd

today = dd.now()
programnumber = 19
regisetr_no = 2117108

date_today = dd.strftime(today,"%d-%B-%Y")

result_file = f"output program_{programnumber}_{regisetr_no}_{date_today}"

open(f"{result_file}.txt",'w').close()

def write_file(content,message=None):
    res = open(f"{result_file}.txt",'a')
    if message is not None:
        print(f"\n{message}\n")
        res.write(f"\n{message}\n\n")
    for ele in content:
        res.write(f"{ele} \t ")
        print(f"{ele}",end=" ")
    res.write("\n")
    print()

file = open('input.txt','r')
line1 = file.readline()
line2 = file.readline()
file.close()

line1_list = line1.strip('\n').split(' ')
line2_list = line2.split(' ')
```

```
line2_list_reversed = line2_list[::-1]
line1_list_reversed = line1_list[::-1]

write_file(line1_list_reversed,"Reverse: ")
write_file(line2_list_reversed)

union_set = set(line1_list).union(set(line2_list))
write_file(union_set,"Union: ")

intersection_set = set(line1_list).intersection(set(line2_list))
write_file(intersection_set,"Intersection: ")
```

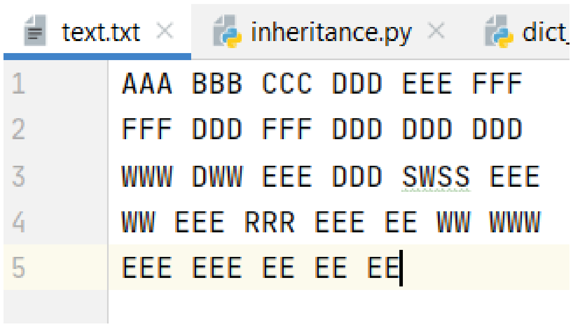


```
Reverse:

EE DD CC BB AA
HH GG FF CC BB

Union:

FF GG DD HH EE CC AA BB

Intersection:

CC BB

Process finished with exit code 0
```

```
"""------------------------------------------------------------------------------------
Lab20.py
20.There is a file with several text messages. Each message is in its own line. Write a Python
program to count the number of lines in the file and the total number of words contained in
those messages. Assume the messages contain only alphabets, and numbers.
Regno: 2117108
07/04/2022
------------------------------------------------------------------------------------"""
file = open('text.txt','r')
content = file.readlines()
number_of_lines = len(content)
word_length = 0

for line in content:
    word_length +=len(line.split(' '))
print(f"The number of lines are : {number_of_lines}")
print(f"The number of words in the file are : {word_length}")
```

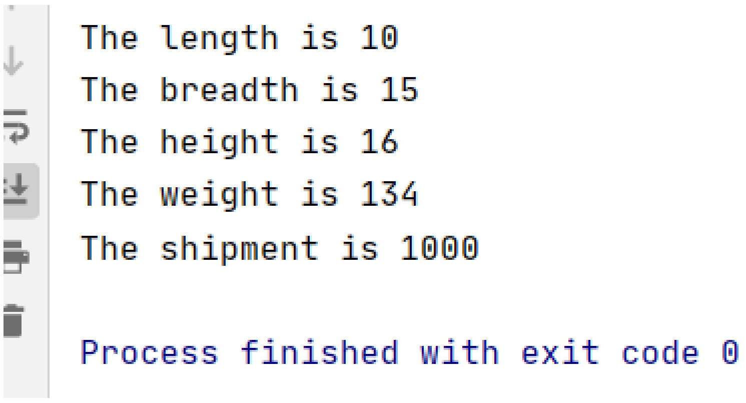


```
The number of lines are : 5
The number of words in the file are : 30

Process finished with exit code 0
```



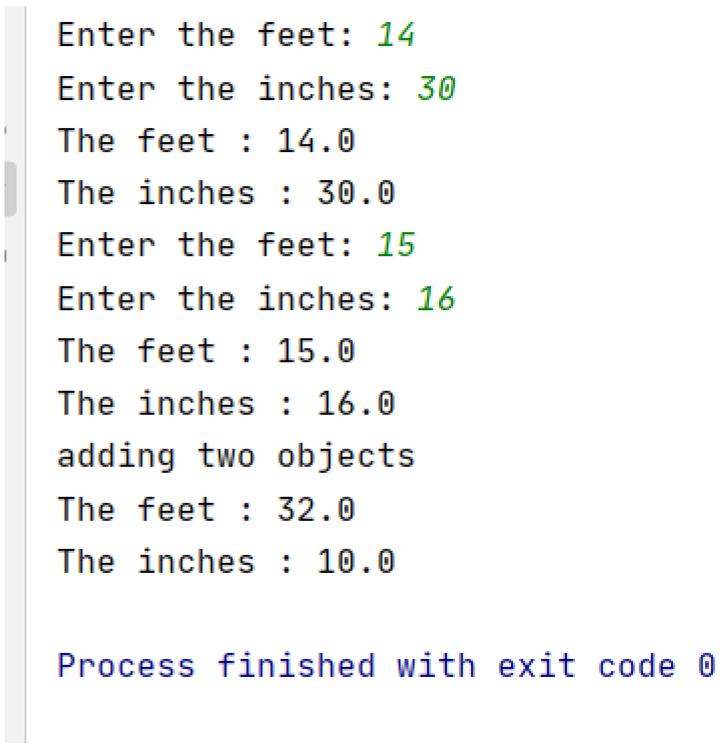
```
AAA BBB CCC DDD EEE FFF
FFF DDD FFF DDD DDD DDD
WWW DWW EEE DDD SWSS EEE
WW EEE RRR EEE EE WW WWW
EEE EEE EE EE EE
```

```
"""-----------------------------------------------------------------------------------
Lab21.py
21.Program to illustrate multilevel inheritance Box (length,breadth,,height) as the super class.
Boxweight (weight) and Boxshipment (cost) as the subclasses. Illustate the use of super
keywords, constructor assign the value not zero
Regno: 2117108
07/04/2022

---------------------------------------------------------------------------------"""
class Box:
    def __init__(self, length, breadth, height):
        self.length = length
        self.breadth = breadth
        self.height = height

    def dislplay(self):
        print(f"The length is {self.length}")
        print(f"The breadth is {self.breadth}")
        print(f"The height is {self.height}")

class BoxWeight(Box):
    def __init__(self, length, breadth, height, weight):
        super(BoxWeight, self).__init__(length, breadth, height)
        self.weight = weight

    def dislplay(self):
        super(BoxWeight, self).dislplay()
        print(f"The weight is {self.weight}")

class BoxShipment(BoxWeight):
    def __init__(self, length, breadth, height, weight, shipment):
        super(BoxShipment, self).__init__(length, breadth, height, weight)
        self.shipment = shipment

    def dislplay(self):
        super(BoxShipment, self).dislplay()
        print(f"The shipment is {self.shipment}")

bs = BoxShipment(10, 15, 16, 134, 1000)
bs.dislplay()
```



```
The length is 10
The breadth is 15
The height is 16
The weight is 134
The shipment is 1000

Process finished with exit code 0
```

```
"""-------------------------------------------------------------------------------
Lab22.py
22.Write a class Distance with instance variables feet and inches. Include necessary
methods. Test the class

Regno: 2117108
08/04/2022
-------------------------------------------------------------------------------"""

class Distance:
    def __init__(self, feet=None, inches=None):
        self.feet: float = feet
        self.inches: float = inches

    def input_data(self):
        self.feet = float(input("Enter the feet: "))
        self.inches = float(input("Enter the inches: "))

    def add_distance(self, distance):
        newfeet = (distance.feet + self.feet) + int((distance.inches + self.inches) / 12)
        newinches = (distance.inches + self.inches) % 12
        return Distance(feet=newfeet, inches=newinches)

    def display(self):
        print(f"The feet : {self.feet}")
        print(f"The inches : {self.inches}")

obj1 = Distance()
obj1.input_data()
obj1.display()

obj2 = Distance()
obj2.input_data()
obj2.display()

print("adding two objects")
newobj = obj1.add_distance(obj2)
newobj.display()
```



```
Enter the feet: 14
Enter the inches: 30
The feet : 14.0
The inches : 30.0
Enter the feet: 15
Enter the inches: 16
The feet : 15.0
The inches : 16.0
adding two objects
The feet : 32.0
The inches : 10.0

Process finished with exit code 0
```

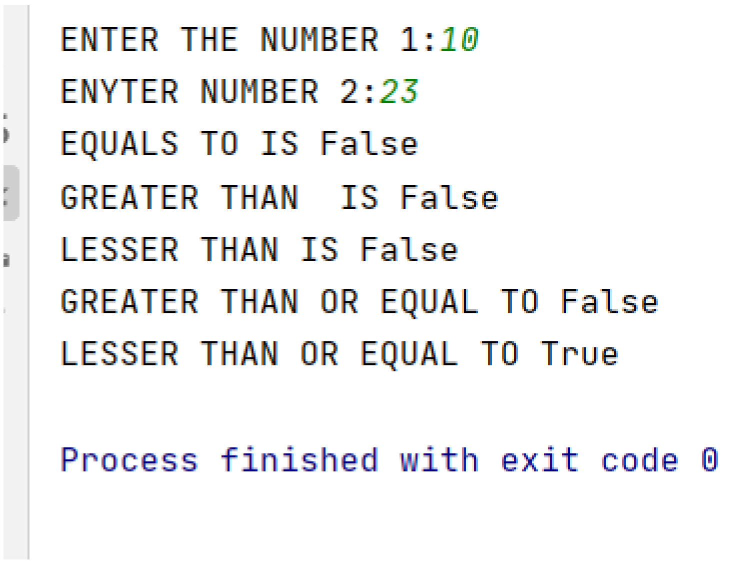
```
'''--------------------------------------------------------------------------------
Lab23.py
23.class Relation with abstract method to implement t basic relational operators (-.,-) on two
integers. Define class number with two data fields (N), N2) which extends class Relation and
illustrate the main class. (hint: use user module all the three program in different fil like
sample car example.)

Regno:2117108
08/04/2022

-------------------------------------------------------------------------------- '''
from module import relation_number as rs

num1 = int(input("ENTER THE NUMBER 1:"))
num2  = int(input("ENYTER NUMBER 2:"))

m1 = rs.Number(num1,num2)

print(f"EQUALS TO IS {m1.equals_to()}")
print(f"GREATER THAN  IS {m1.greater_than()}")
print(f"LESSER THAN IS {m1.lesser_than()}")
print(f"GREATER THAN OR EQUAL TO {m1.greater_than_equals()}")
print(f"LESSER THAN OR EQUAL TO {m1.lesser_than_equals()}")
```

module/abstract_methods_relation.py

```
from abc import ABC, abstractmethod

class example(ABC):
    def __init__(self):
        pass

    @abstractmethod
    def equals_to(self):
        pass

    @abstractmethod
    def greater_than(self):
        pass

    @abstractmethod
    def lesser_than(self):
        pass

    @abstractmethod
    def greater_than_equals(self):
        pass

    @abstractmethod
    def lesser_than_equals(self):
        pass
```

module/relation_number.py

```
from module import abstract_methods_relation

class Number(abstract_methods_relation.example):
    def __init__(self, num1, num2):
        super().__init__()
        self.N1 = num1
        self.N2 = num2

    def equals_to(self):
        return self.N1 == self.N2

    def greater_than(self):
        return self.N1 > self.N2

    def lesser_than(self):
        return self.N1 > self.N2

    def greater_than_equals(self):
        return self.N1 >= self.N2

    def lesser_than_equals(self):
        return self.N1 <= self.N2
```



```
ENTER THE NUMBER 1:10
ENYTER NUMBER 2:23
EQUALS TO IS False
GREATER THAN  IS False
LESSER THAN IS False
GREATER THAN OR EQUAL TO False
LESSER THAN OR EQUAL TO True

Process finished with exit code 0
```

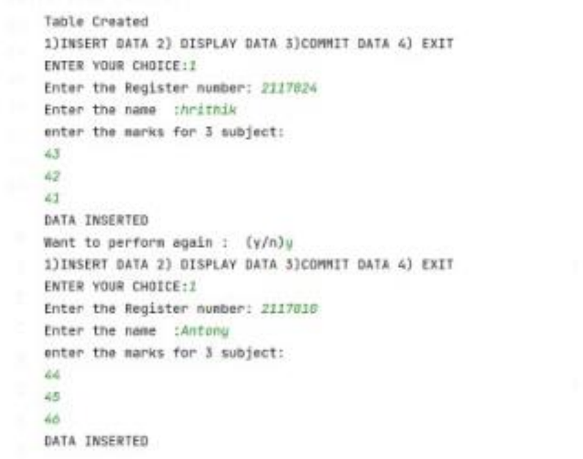
#____
Lab 24.py
 Write a python program to add few customer details into the database and retrieve the information and print in systematic manner .
Regno: 2117108
Date: 08/04/2022
#____

```
import sqlite3 as db

conn = db.Connection('customer.db')
cursor = conn.cursor()
cursor.execute("create table if not exists customer(id integer primary key,name text,salary
integer,address text)")



class Customer:
    def __init__(self):
        pass
    def insert_data(self, cust_id, cust_name, cust_sal, cust_address):
        if cursor.execute(f"insert into customer
values({cust_id},'{cust_name}',{cust_sal},'{cust_address}')"):
          print("data inserted")
        else:
          print("data insertion failed")
    def print_data(self, cust_id=None):
        if cust_id is not None:
          query = f"select * from customer where id={cust_id}"
        else:
          query = "select * from customer"
          data = cursor.execute(query)
          data = data.fetchall()
          print("id\tname\tsalary\taddress")
          for row in data:
            print(f"{row[0]}\t{row[1]}\t{row[2]}\t{row[3]}")
    def update_data(self):
         cust_id = int(input("Enter the customer Id whose data you wish to update: "))
         if cursor.execute(f"select * from customer where id={cust_id}"):
            name = input("Enter the Name: ")
            salary = int(input("Enter the salary: "))
            address = input("Enter the Address: ")
            if cursor.execute(f"update customer set
name='{name}',salary={salary},address='{address}' where id={cust_id}"):
               print("Data successfully updated")
            else:
               print("Data updation failed")
         else:
            print(f"no data found for custoner id {cust_id}")
    def delete_data(self):
      cust_id = int(input("Enter the custid you wish to delete: "))
      if cursor.execute(f"delete from customer where id={cust_id}"):
        print("Data successfully deleted")
      else:
        print("Data deletion failed")

while True:
   c = Customer()
   choice = int(input("Enter the choice\n1. Insert\t2. Display\t3.Display
specific\t4.update\t5.Delete\t6.Exit\nEnter your choice: "))
   if choice == 1:
      c_id = int(input("Enter the customer id: "))
      name = input("Enter the name: ")
      salary = int(input("Enter the salary: "))
      address = input("Enter the address: ")
      c.insert_data(cust_id=c_id, cust_name=name, cust_sal=salary,cust_address=address)
        # insert_data(cust_id=101,
cust_name='namita',cust_sal=20000,cust_address='mangalore')
   elif choice == 2:
         c.print_data()
   elif choice == 3:
         cust_id = int(input("Enter the customer number: "))
         c.print_data(cust_id)
   elif choice == 4:
         c.update_data()
   elif choice == 5:
         c.delete_data()
   elif choice == 6:
         conn.close()
         break
   else:
        print("Invalid choice.")
```

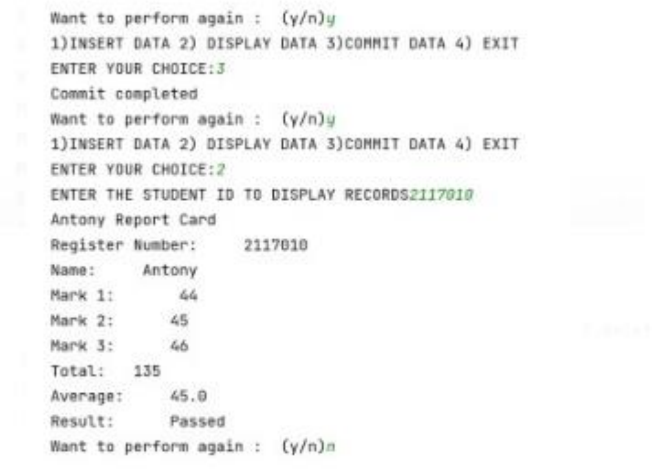
```
D:\python\venv\Scripts\python.exe D:/python/lab24.py
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 1
Enter the customer id: 101
Enter the name: nena
Enter the salary: 300000
Enter the address: karwar
data inserted
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 1
Enter the customer id: 201
Enter the name: maya
Enter the salary: 567000
Enter the address: kumta
data inserted
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 2
id  name    salary  address
101 nena    300000  karwar
201 maya    567000  kumta
```

```
Enter your choice: 5
Enter the custid you wish to delete: 201
Data successfully deleted
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 2
id  name    salary  address
101 nena    300000  karwar
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 3
Enter the customer number: 101
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 4
Enter the customer Id whose data you wish to update: 101
Enter the Name: rahul
Enter the salary: 450000
Enter the Address: bhatkal
Data successfully updated
Enter the choice
1. Insert   2. Display  3.Display specific  4.update    5.Delete    6.Exit
Enter your choice: 6

Process finished with exit code 0
```

```
'''--------------------------------------------------------------

LAB25.py

Write a python application to generate student report card enter all the details of the student
required. Calculate the marks, average,and resuls, then update the information
accordingly.Write a python application to generate student report card enter all the details of
the student required. Calculate the marks, average,and resuls, then update the information
accordingly.

RegNo: 2117108

08-04-2022

--------------------------------------------------------------'''

import sqlite3 as s

conn = s.connect('Student.db')
cursor = conn.cursor()
cursor.execute("DROP TABLE  IF EXISTS Student")
cursor.execute("create table  Student(id integer primary key,name text,mark1 integer,mark2 integer,"
            "mark3 integer,total integer ,average real,result text)")

print("Table Created")
class Student:

    def Insert(self):
        self.regNo = int(input("Enter the Register number"))
        self.name = input("Enter the name \t :")
        print("enter the marks for 3 subject")
        self.mark1 = int(input())
        self.mark2 = int(input())
        self.mark3 = int(input())
        self.total = self.mark1 + self.mark2 + self.mark3
        self.avg = self.total / 3

        if self.mark1 < 35 or self.mark2 < 35 or self.mark2 < 35:
            self.result = "failed"
        elif self.avg >= 90:
            result = "Distinction"
        elif self.avg >= 70:
            self.result = "First"
        elif self.avg >= 50:
            self.result = "Second"
        elif self.avg >= 35:
            self.result = "Passed"
        if cursor.execute("insert into Student values(?,?,?,?,?,?,?,?)",
                         (self.regNo, self.name, self.mark1, self.mark2, self.mark3, self.total, self.avg,
                          self.result)):
            print("DATA INSERTED")
        else:
            print("NOT INSERTED DATA")

    def Display(srlf):
        id = int(input("ENTER THE STUDENT ID TO DISPLAY RECORDS"))

        cursor.execute(f"select * from Student where id = {id}")
        data = cursor.fetchone()
        if (data is not None):
            print(f"{data[1]} Report Card")
            print("Register Number: \t", data[0])
            print("Name: \t ", data[1])
            print("Mark 1: \t ", data[2])
            print("Mark 2: \t", data[3])
            print("Mark 3: \t", data[4])
            print("Total: \t", data[5])
            print("Average: \t", data[6])
            print("Result: \t", data[7])
        else:
            print("STUDENT DATA IS NOT FOUND ")

S = Student()
y = "y"
while y == 'y':

    print("1)INSERT DATA\t 2) DISPLAY DATA 3)COMMIT DATA 4) EXIT")
    c = int(input("ENTER YOUR CHOICE:"))
    if c == 1:
        S.Insert()
    elif c == 2:
        S.Display()
    elif c == 3:
        conn.commit()
        print("data Committed")
    elif c == 4:
        exit()
    else:
        print("WRONG CHOICE")

    y = input("Want to perform again :\t (y/n)")
```



```
Table Created
1)INSERT DATA 2) DISPLAY DATA 3)COMMIT DATA 4) EXIT
ENTER YOUR CHOICE:1
Enter the Register number: 2117024
Enter the name  :hrithik
enter the marks for 3 subject:
43
42
41
DATA INSERTED
Want to perform again :  (y/n)y
1)INSERT DATA 2) DISPLAY DATA 3)COMMIT DATA 4) EXIT
ENTER YOUR CHOICE:1
Enter the Register number: 2117010
Enter the name  :Antony
enter the marks for 3 subject:
44
45
46
DATA INSERTED
```



```
Want to perform again :  (y/n)y
1)INSERT DATA 2) DISPLAY DATA 3)COMMIT DATA 4) EXIT
ENTER YOUR CHOICE:3
Commit completed
Want to perform again :  (y/n)y
1)INSERT DATA 2) DISPLAY DATA 3)COMMIT DATA 4) EXIT
ENTER YOUR CHOICE:2
ENTER THE STUDENT ID TO DISPLAY RECORDS2117010
Antony Report Card
Register Number:     2117010
Name:     Antony
Mark 1:       44
Mark 2:      45
Mark 3:      46
Total:   135
Average:     45.0
Result:      Passed
Want to perform again :  (y/n)n
```