UPES

UNIVERSITY OF THE FUTURE

PYTHON PROGRAMMING LABS
SMALL BUSINESS MANAGEMENT SYSTEM
(B.TECH CSE 2023-27)

SUBMITTED BY:

NAME: - DAKSH DEVRANI

SAP ID: - 500123950

ROLL NO: - R2142230673

BATCH: - 22

SUBMITTED TO:

NAME: - DR. SWATI RASTOGI

Lab – 1

1) Install Python and understand difference between scripting and interactive modes in IDLE.

Ans: To install Python, you can follow these general steps:

- Visit the official Python website: https://www.python.org/
- Go to the "Downloads" section.
- Choose the appropriate installer for your operating system (Windows, macOS, or Linux).
- Download the installer and run it.
- Follow the installation wizard instructions, ensuring you select the option to add Python to your PATH environment variable during installation.
- Once Python is installed, you can launch IDLE (Integrated Development and Learning Environment), which is a Python-specific Integrated Development Environment (IDE) that comes bundled with the Python installer.

Now, let's understand the difference between scripting mode and interactive mode in IDLE:

Scripting Mode:

- In scripting mode, you write your Python code in a script file (with a .py extension) using a text editor or an IDE like IDLE.
- You save the script file on your computer's filesystem.
- You execute the script file using the Python interpreter.
- The Python interpreter reads the script file from start to end, executing each line of code sequentially.
- Scripting mode is suitable for writing larger programs or scripts that you intend to reuse or distribute.

Interactive Mode:

- In interactive mode, you directly interact with the Python interpreter in real-time.
- You launch IDLE or open a terminal/console, and then type Python code directly into the interpreter prompt (>>>).
- The Python interpreter executes each line of code immediately after you press Enter.
- Interactive mode is useful for experimenting with Python syntax, testing small code snippets, or performing quick calculations.
- It provides instant feedback and allows you to quickly see the results of your code without needing to save and run a separate script file.

2) Write Python programs to print strings in the given manner:

a) Hello Everyone !!!

b) Hello

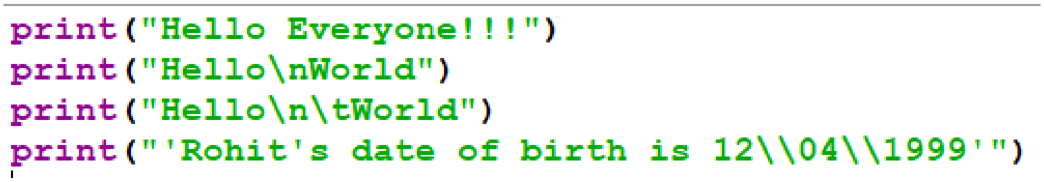
World

c) Hello

World

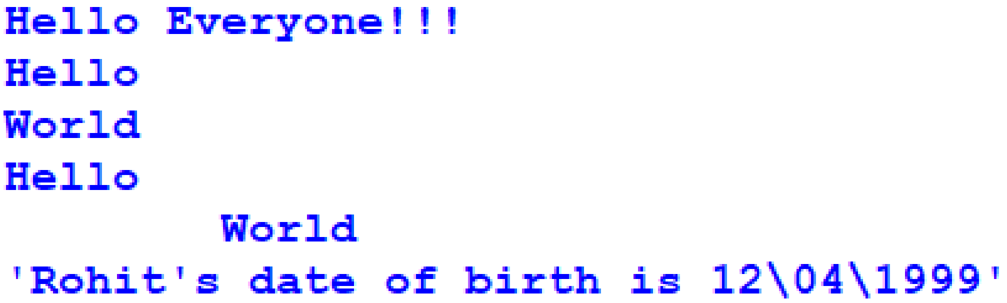
d) ‘ Rohit’ s date of birth is 12\05\1999’

Ans:



```
print("Hello Everyone!!!")
print("Hello\nWorld")
print("Hello\n\tWorld")
print("'Rohit's date of birth is 12\\04\\1999'")
```

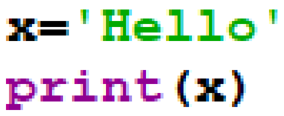
Output:



```
Hello Everyone!!!
Hello
World
Hello
        World
'Rohit's date of birth is 12\04\1999'
```

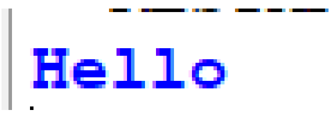
3) Declare a string variable called x and assign it the value “Hello”, Print out the value of x

Ans:



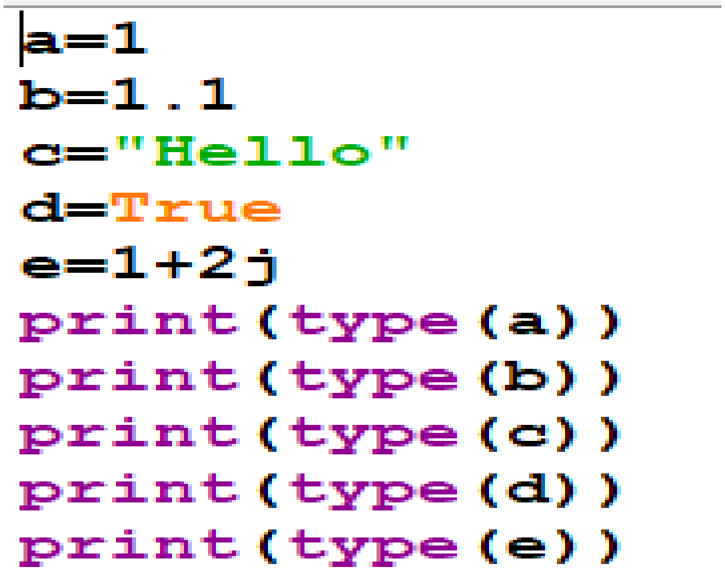
```
x='Hello'
print(x)
```

Output:



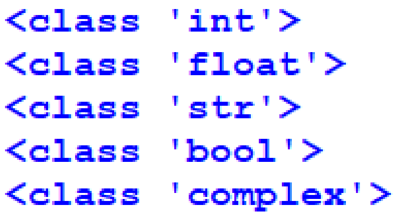
```
Hello
```

4) Take different data types and print values using print function.
Ans:



```
a=1
b=1.1
c="Hello"
d=True
e=1+2j
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
```

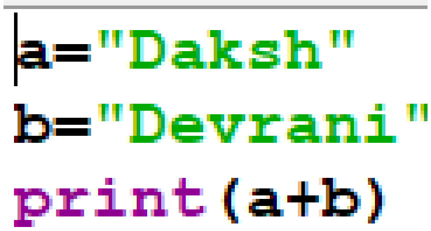
Output:



```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'complex'>
```

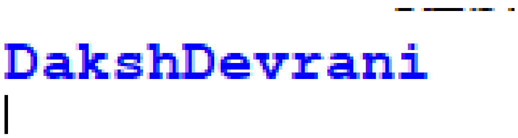
5) Take two variables, a and b. Assign your first name and last name. Print your Name after adding your First name and Last name together

Ans:



```
a="Daksh"
b="Devrani"
print(a+b)
```

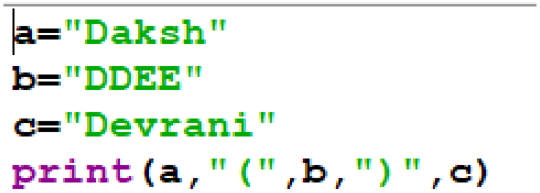
Output:



```
DakshDevrani
```

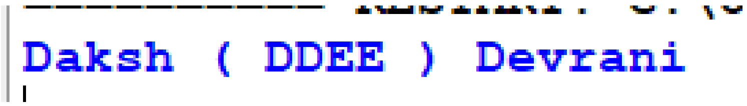
6) Declare three variables, consisting of your first name, your last name and Nickname. Write a program that prints out your first name, then your nickname in parenthesis and then your last name. Example output : George (woody) Washington.

Ans:



```
a="Daksh"
b="DDEE"
c="Devrani"
print(a,"(",b,")",c)
```

Output:



```
Daksh ( DDEE ) Devrani
```

7) Declare and assign values to suitable variables and print in the following way :

NAME : NIKUNJ BANSAL

SAP ID : 500069944

DATE OF BIRTH : 13 Oct 1999

ADDRESS : UPES

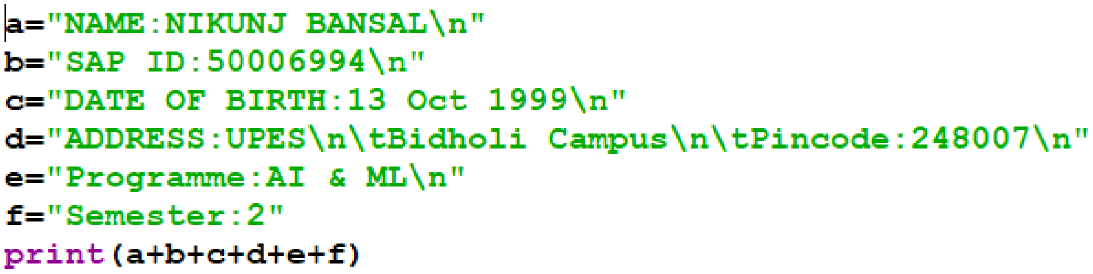
Bidholi Campus

Pincode : 248007

Programme : AI & ML

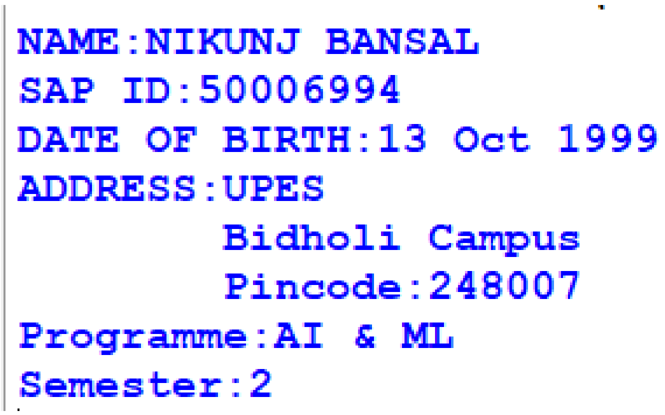
Semester : 2

Ans:



```
a="NAME:NIKUNJ BANSAL\n"
b="SAP ID:50006994\n"
c="DATE OF BIRTH:13 Oct 1999\n"
d="ADDRESS:UPES\n\tBidholi Campus\n\tPincode:248007\n"
e="Programme:AI & ML\n"
f="Semester:2"
print(a+b+c+d+e+f)
```

Output:

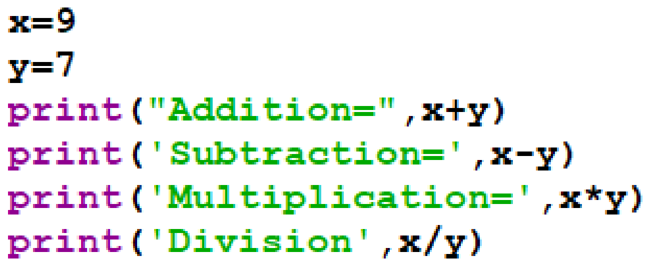


```
NAME:NIKUNJ BANSAL
SAP ID:50006994
DATE OF BIRTH:13 Oct 1999
ADDRESS:UPES
        Bidholi Campus
        Pincode:248007
Programme:AI & ML
Semester:2
```

Lab – 2

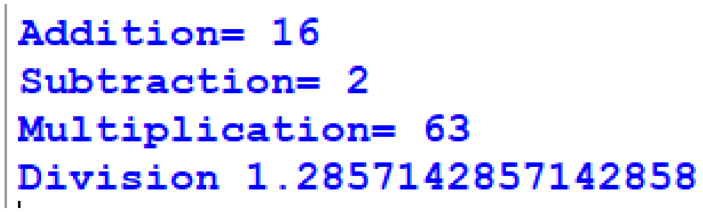
1) Declare these variables (x, y and z) as integers. Assign a value of 9 to x, Assign a value of 7 to y, perform addition, multiplication, division and subtraction on these two variables and Print out the result.

Ans:



```
x=9
y=7
print("Addition=",x+y)
print('Subtraction=',x-y)
print('Multiplication=',x*y)
print('Division',x/y)
```

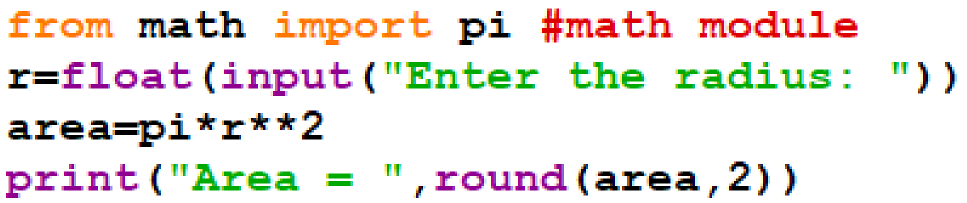
Output:



```
Addition= 16
Subtraction= 2
Multiplication= 63
Division 1.2857142857142858
```

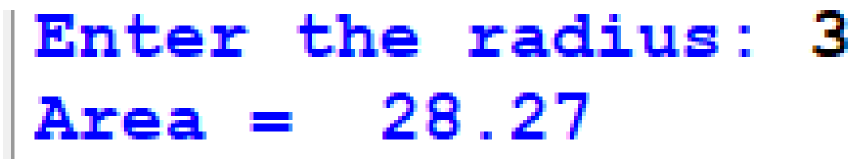
2) Write a Program where the radius is taken as input to compute the area of a circle.

Ans:



```
from math import pi #math module
r=float(input("Enter the radius: "))
area=pi*r**2
print("Area = ",round(area,2))
```

Output:



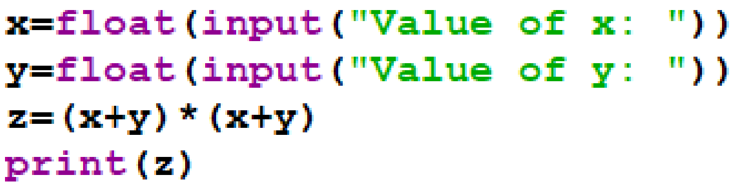
```
Enter the radius: 3
Area =  28.27
```

3) Write a Python program to solve (x+y)*(x+y)

a. Test data : x = 4 , y = 3

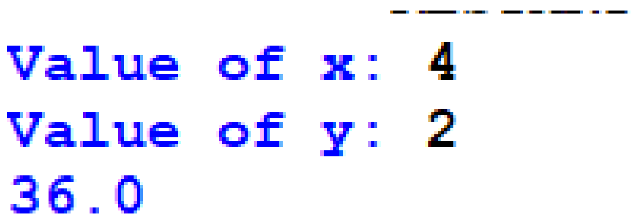
b. Expected output: 49

Ans:



```
x=float(input("Value of x: "))
y=float(input("Value of y: "))
z=(x+y)*(x+y)
print(z)
```

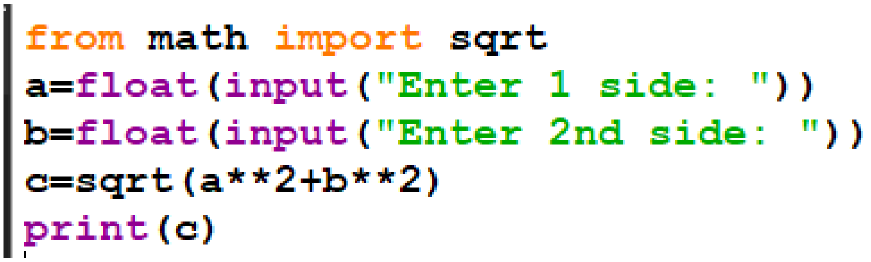
Output:



```
Value of x: 4
Value of y: 2
36.0
```

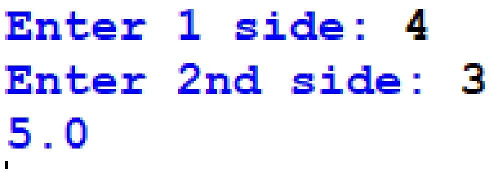
4)) Write a program to compute the length of the hypotenuse (c) of a right triangle using Pythagoras theorem.

Ans:



```
from math import sqrt
a=float(input("Enter 1 side: "))
b=float(input("Enter 2nd side: "))
c=sqrt(a**2+b**2)
print(c)
```

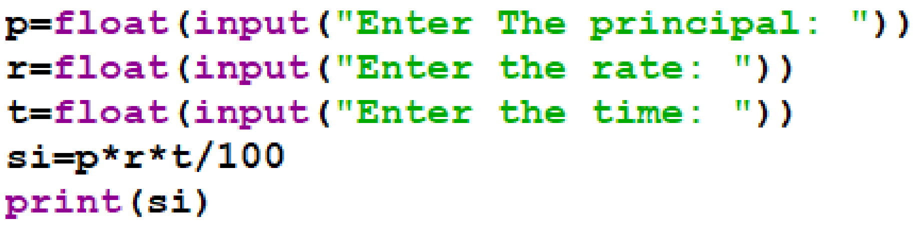
Output:



```
Enter 1 side: 4
Enter 2nd side: 3
5.0
```

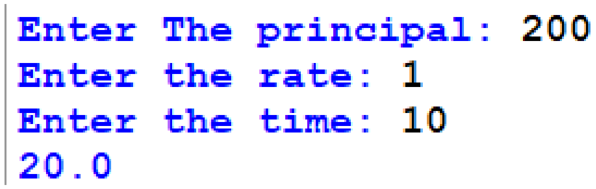
5) Write a program to find simple interest.

Ans:



```
p=float(input("Enter The principal:  "))
r=float(input("Enter the rate:  "))
t=float(input("Enter the time:  "))
si=p*r*t/100
print(si)
```

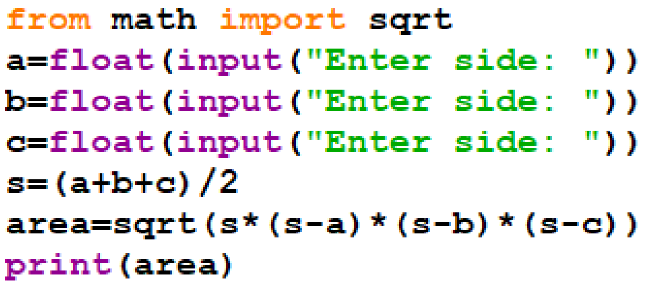
Output:



```
Enter The principal: 200
Enter the rate: 1
Enter the time: 10
20.0
```

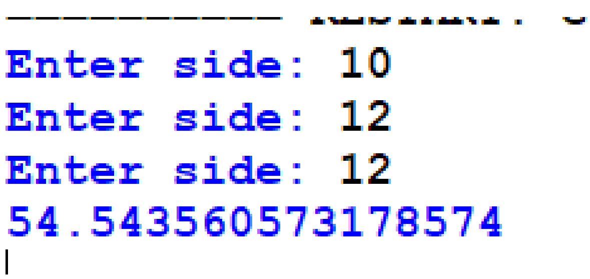
6) Write a program to find area of triangle when length of sides are given

Ans:



```
from math import sqrt
a=float(input("Enter side:  "))
b=float(input("Enter side:  "))
c=float(input("Enter side:  "))
s=(a+b+c)/2
area=sqrt(s*(s-a)*(s-b)*(s-c))
print(area)
```

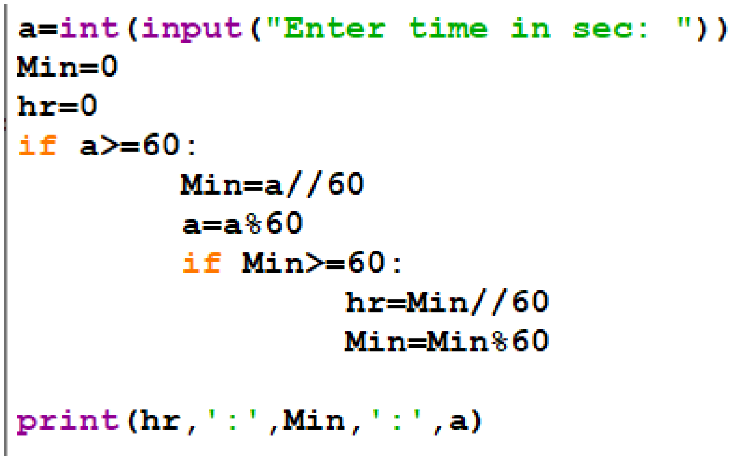
Output:



```
Enter side: 10
Enter side: 12
Enter side: 12
54.543560573178574
```

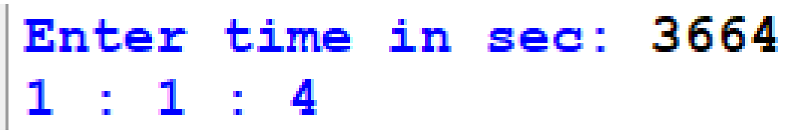
7) Write a program to convert given seconds into hours, minutes and remaining seconds.

Ans:



```
a=int(input("Enter time in sec: "))
Min=0
hr=0
if a>=60:
        Min=a//60
        a=a%60
        if Min>=60:
                hr=Min//60
                Min=Min%60

print(hr,':',Min,':',a)
```

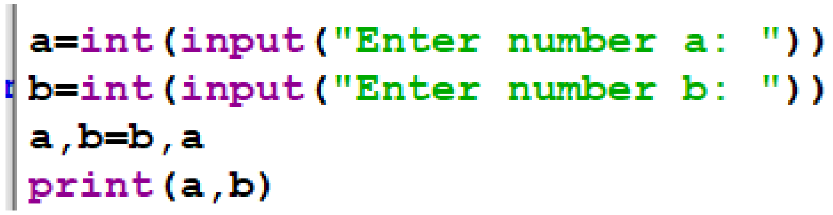
Output:



```
Enter time in sec: 3664
1 : 1 : 4
```

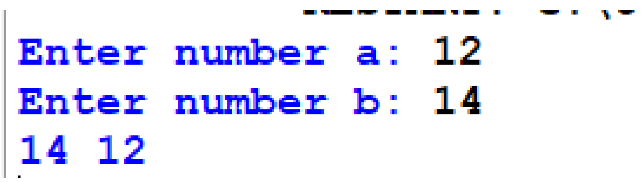
8) Write a program to swap two numbers without taking additional variable

Ans:



```
a=int(input("Enter number a: "))
b=int(input("Enter number b: "))
a,b=b,a
print(a,b)
```

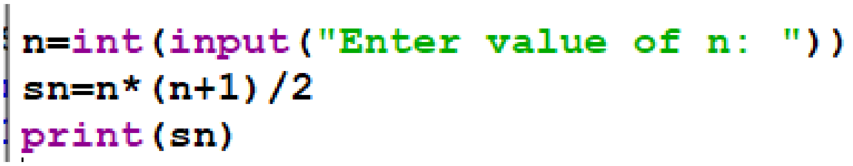
Output:



```
Enter number a: 12
Enter number b: 14
14 12
```

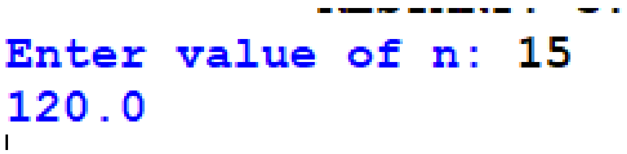
9) Write a program to find sum of first n natural numbers.

Ans:



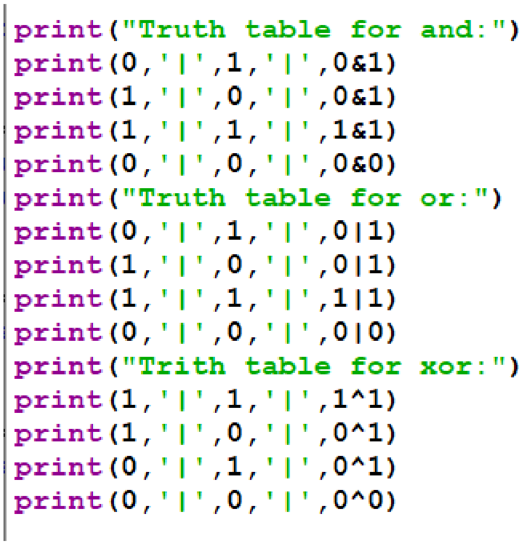
```
n=int(input("Enter value of n:  "))
sn=n*(n+1)/2
print(sn)
```

Output:



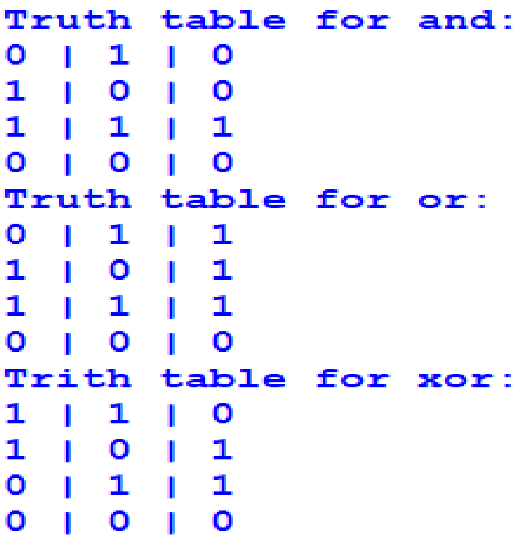
```
Enter value of n: 15
120.0
```

10) Write a program to print truth table for bitwise operators(& , | and ^ operators)



```
print("Truth table for and:")
print(0,'|',1,'|',0&1)
print(1,'|',0,'|',0&1)
print(1,'|',1,'|',1&1)
print(0,'|',0,'|',0&0)
print("Truth table for or:")
print(0,'|',1,'|',0|1)
print(1,'|',0,'|',0|1)
print(1,'|',1,'|',1|1)
print(0,'|',0,'|',0|0)
print("Trith table for xor:")
print(1,'|',1,'|',1^1)
print(1,'|',0,'|',0^1)
print(0,'|',1,'|',0^1)
print(0,'|',0,'|',0^0)
```

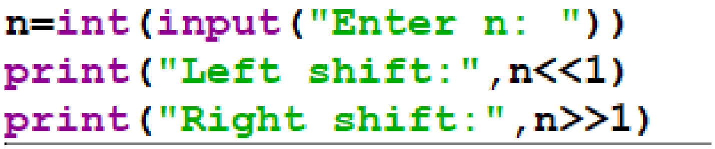
Output:



```
Truth table for and:
0 | 1 | 0
1 | 0 | 0
1 | 1 | 1
0 | 0 | 0
Truth table for or:
0 | 1 | 1
1 | 0 | 1
1 | 1 | 1
0 | 0 | 0
Trith table for xor:
1 | 1 | 0
1 | 0 | 1
0 | 1 | 1
0 | 0 | 0
```

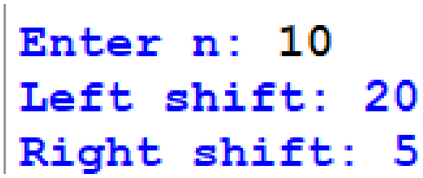
11) Write a program to find left shift and right shift values of a given number. (use bitwise shift)

Ans:



```
n=int(input("Enter n:  "))
print("Left shift:",n<<1)
print("Right shift:",n>>1)
```

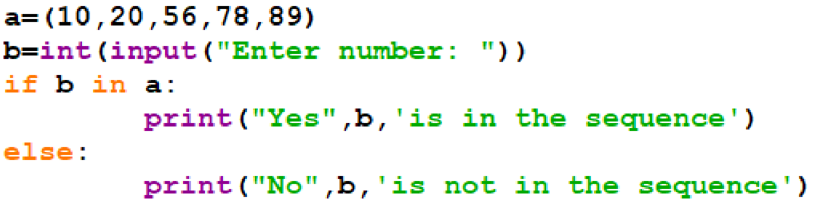
Output:



```
Enter n: 10
Left shift: 20
Right shift: 5
```

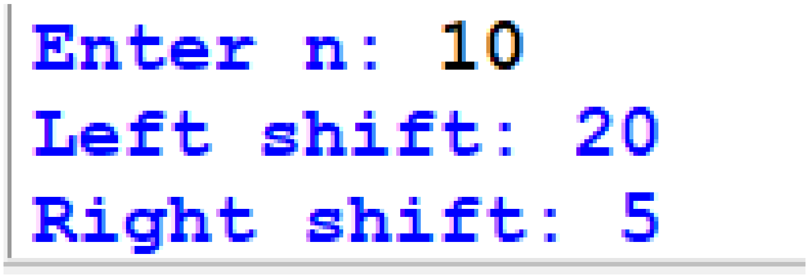
12)) Using membership operator find whether a given number is in sequence (10,20,56,78,89)

Ans:



```
a=(10,20,56,78,89)
b=int(input("Enter number: "))
if b in a:
        print("Yes",b,'is in the sequence')
else:
        print("No",b,'is not in the sequence')
```

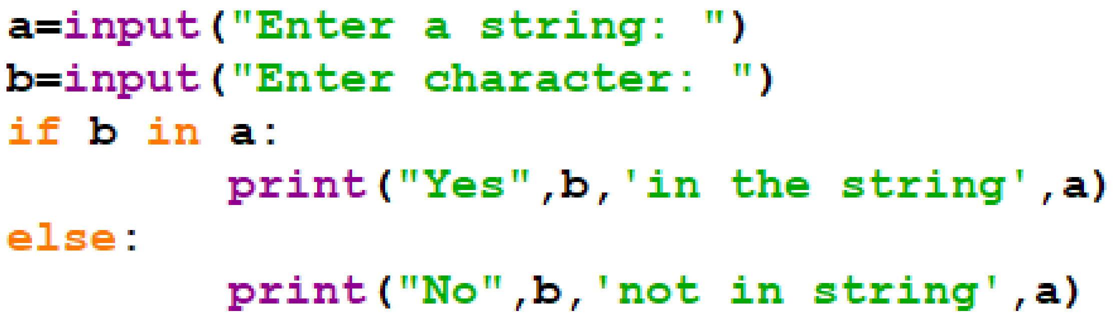
Output:



```
Enter n: 10
Left shift: 20
Right shift: 5
```

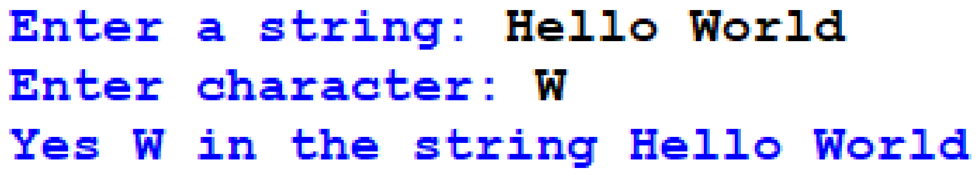
13) Using membership operator find whether a given character is in a string.

Ans:



```
a=input("Enter a string: ")
b=input("Enter character: ")
if b in a:
        print("Yes",b,'in the string',a)
else:
        print("No",b,'not in string',a)
```

Output:

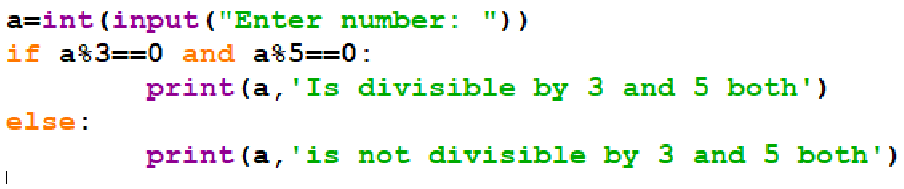


```
Enter a string: Hello World
Enter character: W
Yes W in the string Hello World
```

Lab – 3

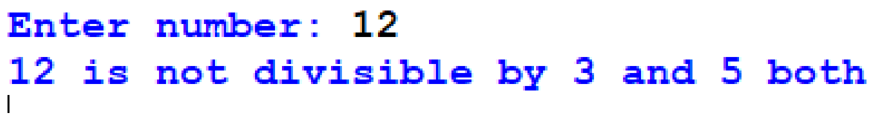
1) Check whether given number is divisible by 3 and 5 both.

Ans:



```
a=int(input("Enter number: "))
if a%3==0 and a%5==0:
        print(a,'Is divisible by 3 and 5 both')
else:
        print(a,'is not divisible by 3 and 5 both')
```

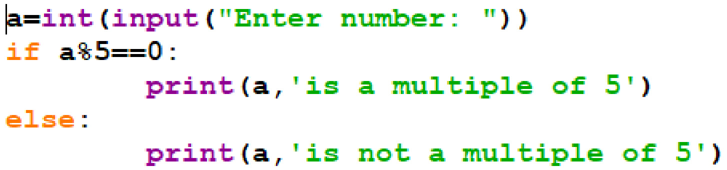
Output:



```
Enter number: 12
12 is not divisible by 3 and 5 both
```

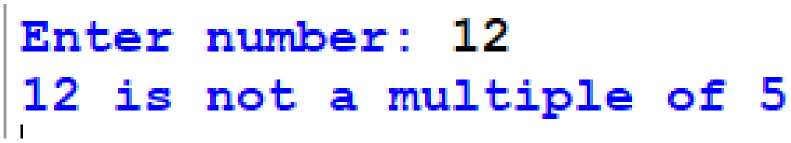
2) Check whether a given number is multiple of five or not.

Ans:



```
a=int(input("Enter number: "))
if a%5==0:
        print(a,'is a multiple of 5')
else:
        print(a,'is not a multiple of 5')
```

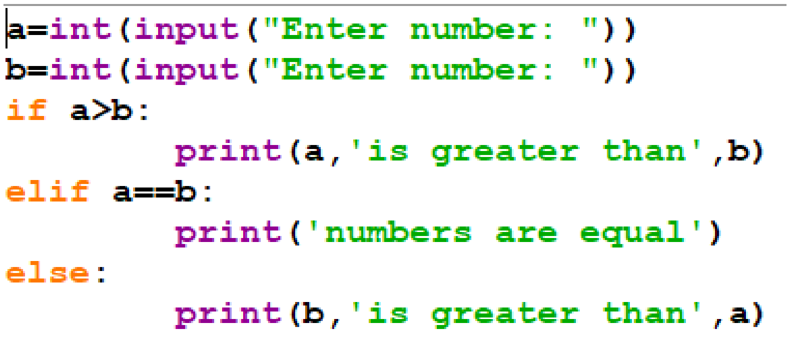
Output:



```
Enter number: 12
12 is not a multiple of 5
```

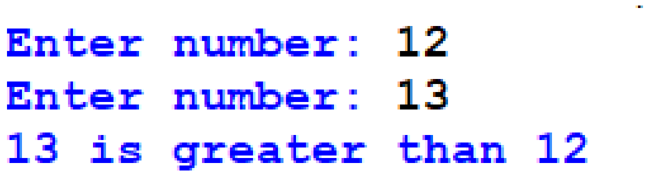
3) Find the greatest among two numbers. If numbers are equal than print “numbers are equal”.

Ans:



```
a=int(input("Enter number: "))
b=int(input("Enter number: "))
if a>b:
        print(a,'is greater than',b)
elif a==b:
        print('numbers are equal')
else:
        print(b,'is greater than',a)
```

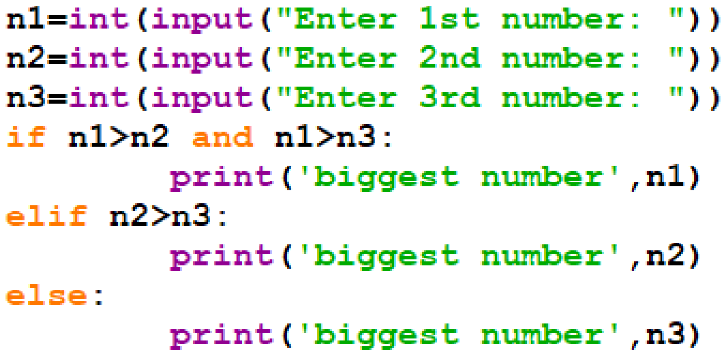
Output:



```
Enter number: 12
Enter number: 13
13 is greater than 12
```

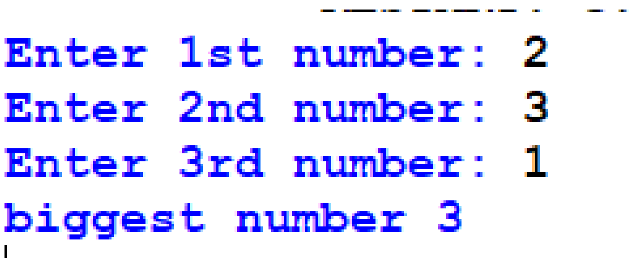
4) Find the greatest among three numbers assuming no two values are same.

Ans:



```
n1=int(input("Enter 1st number: "))
n2=int(input("Enter 2nd number: "))
n3=int(input("Enter 3rd number: "))
if n1>n2 and n1>n3:
        print('biggest number',n1)
elif n2>n3:
        print('biggest number',n2)
else:
        print('biggest number',n3)
```

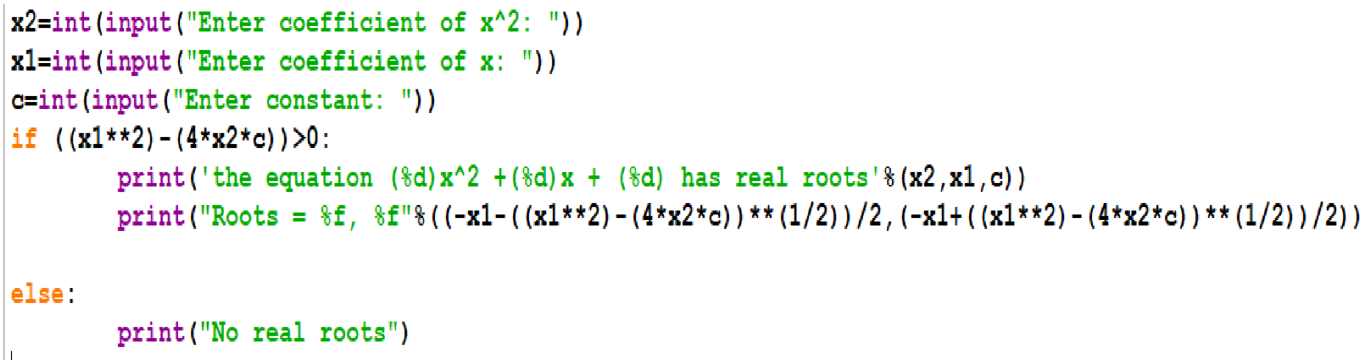
Output:



```
Enter 1st number: 2
Enter 2nd number: 3
Enter 3rd number: 1
biggest number 3
```

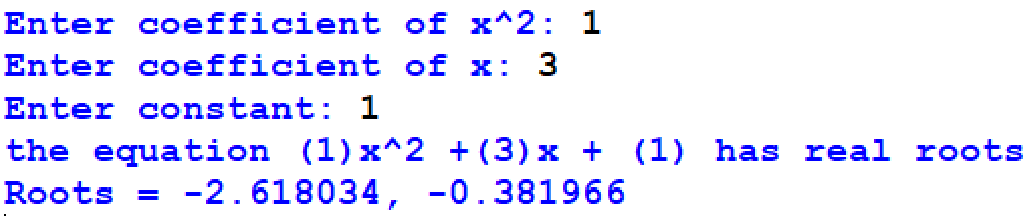
5) Check whether the quadratic equation has real roots or imaginary roots. Display the roots.

Ans:



```
x2=int(input("Enter coefficient of x^2: "))
x1=int(input("Enter coefficient of x: "))
c=int(input("Enter constant: "))
if ((x1**2)-(4*x2*c))>0:
        print('the equation (%d)x^2 +(%d)x + (%d) has real roots'%(x2,x1,c))
        print("Roots = %f, %f"%((-x1-((x1**2)-(4*x2*c))**(1/2))/2,(-x1+((x1**2)-(4*x2*c))**(1/2))/2))

else:
        print("No real roots")
```

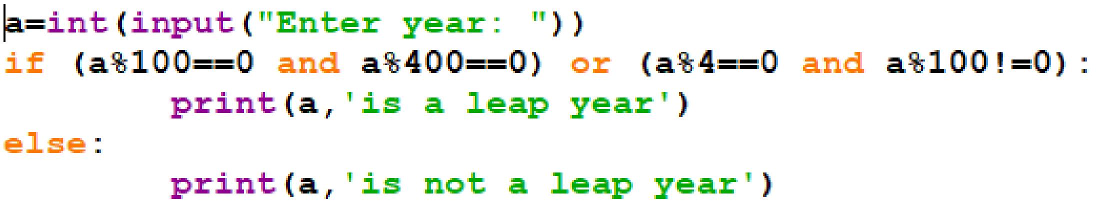
Output:



```
Enter coefficient of x^2: 1
Enter coefficient of x: 3
Enter constant: 1
the equation (1)x^2 +(3)x + (1) has real roots
Roots = -2.618034, -0.381966
```

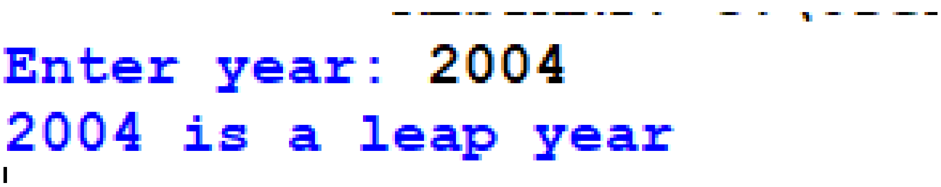
6) Find whether a given year is a leap year or not.

Ans:



```
a=int(input("Enter year: "))
if (a%100==0 and a%400==0) or (a%4==0 and a%100!=0):
        print(a,'is a leap year')
else:
        print(a,'is not a leap year')
```

Output:



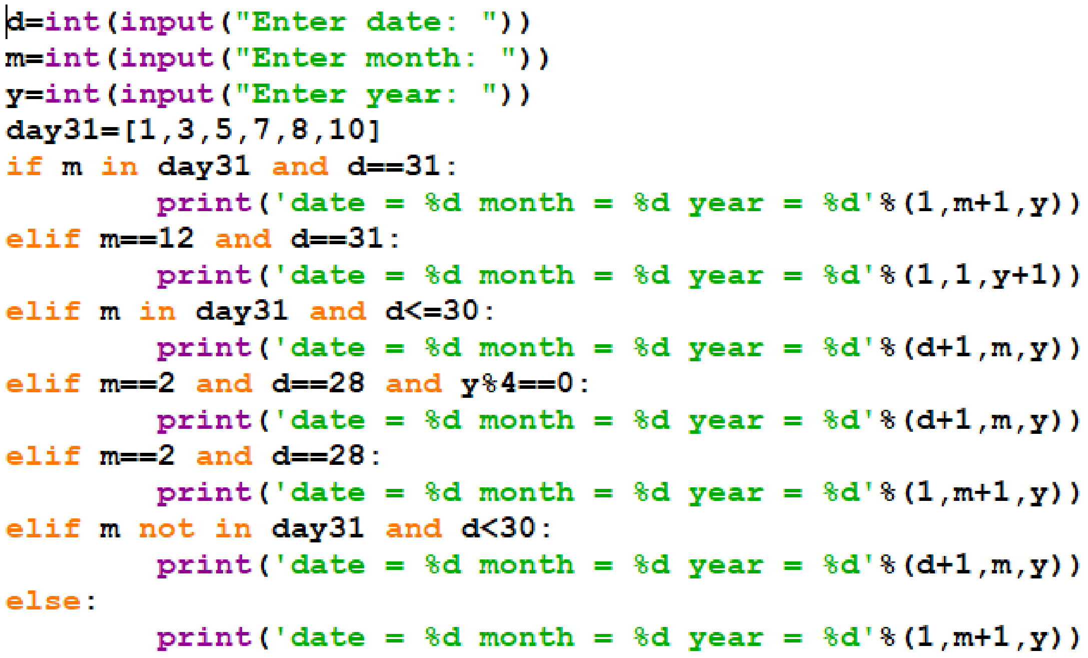
```
Enter year: 2004
2004 is a leap year
```

7) Write a program which takes any date as input and display next date of the calendar. e.g.

1. I/P: day=20 month=9 year=2005

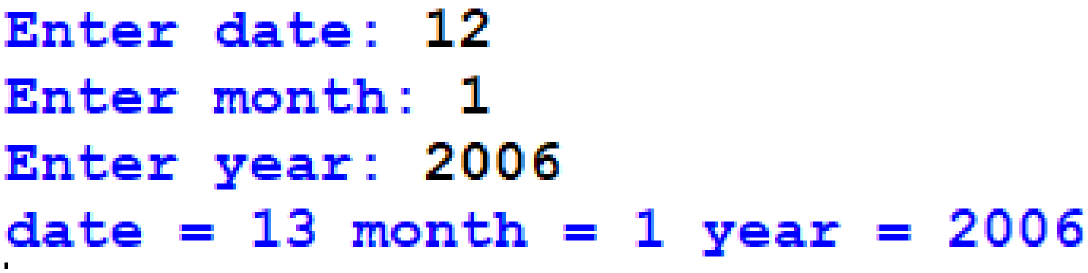
2. O/P: day=21 month=9 year 2005

Ans:



```
d=int(input("Enter date: "))
m=int(input("Enter month: "))
y=int(input("Enter year: "))
day31=[1,3,5,7,8,10]
if m in day31 and d==31:
        print('date = %d month = %d year = %d'%(1,m+1,y))
elif m==12 and d==31:
        print('date = %d month = %d year = %d'%(1,1,y+1))
elif m in day31 and d<=30:
        print('date = %d month = %d year = %d'%(d+1,m,y))
elif m==2 and d==28 and y%4==0:
        print('date = %d month = %d year = %d'%(d+1,m,y))
elif m==2 and d==28:
        print('date = %d month = %d year = %d'%(1,m+1,y))
elif m not in day31 and d<30:
        print('date = %d month = %d year = %d'%(d+1,m,y))
else:
        print('date = %d month = %d year = %d'%(1,m+1,y))
```

Output:



```
Enter date: 12
Enter month: 1
Enter year: 2006
date = 13 month = 1 year = 2006
```

8) Print the grade sheet of a student for the given range of CGPA. Scan marks of five subjects and calculate the

percentage.

CGPA=percentage/10

CGPA range:

0 to 3.4 -> F

3.5 to 5.0->C+

5.1 to 6->B

6.1 to 7-> B+

7.1 to 8-> A

8.1 to 9->A+

9.1 to 10-> O (Outstanding)

Sample Gradesheet

Name: Rohit Sharma

Roll Number: R17234512	**SAPID: 50005673**
Sem: 1	**Course: B.Tech. CSE AI&ML**

Subject name: Marks

PDS: 70

Python: 80

Chemistry: 90

English: 60

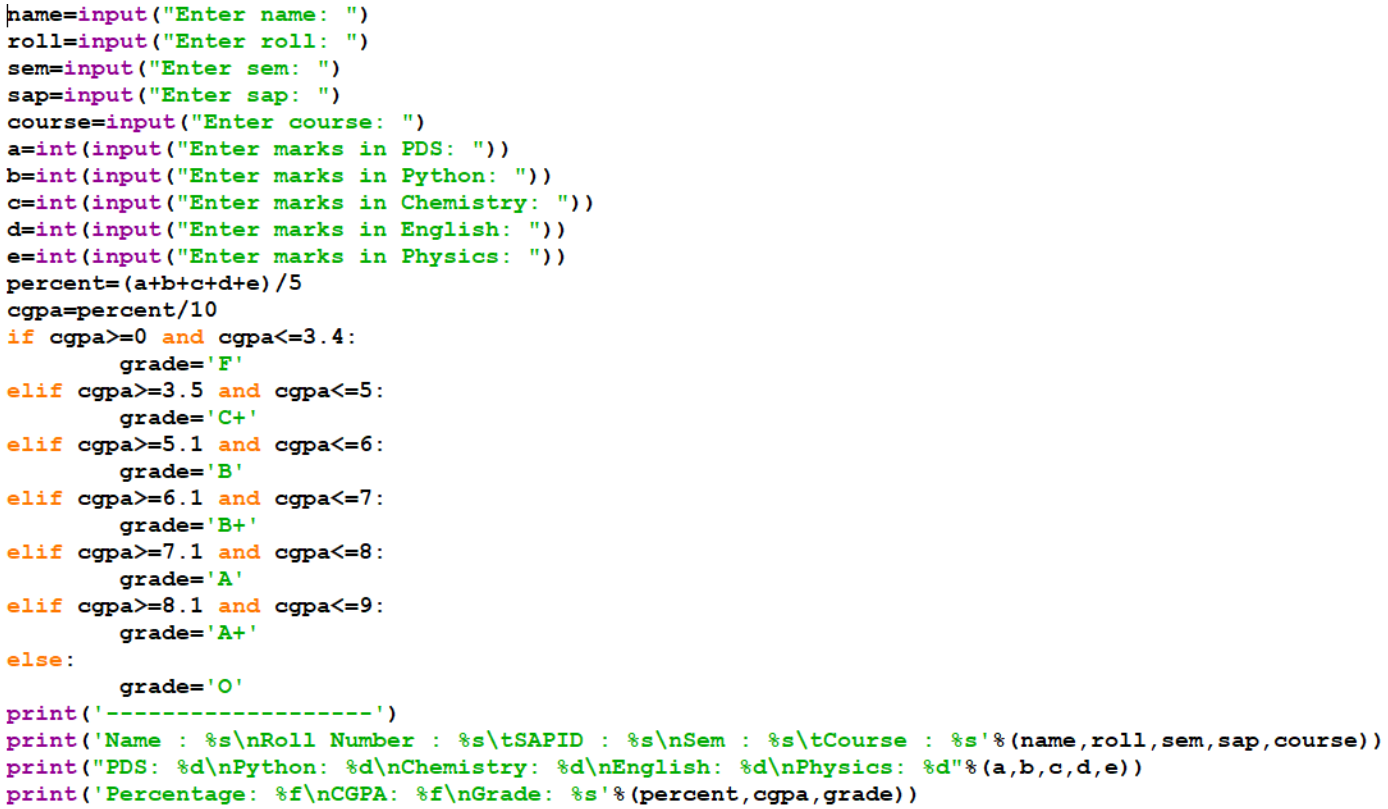
Physics: 50

Percentage: 70%

CGPA:7.0

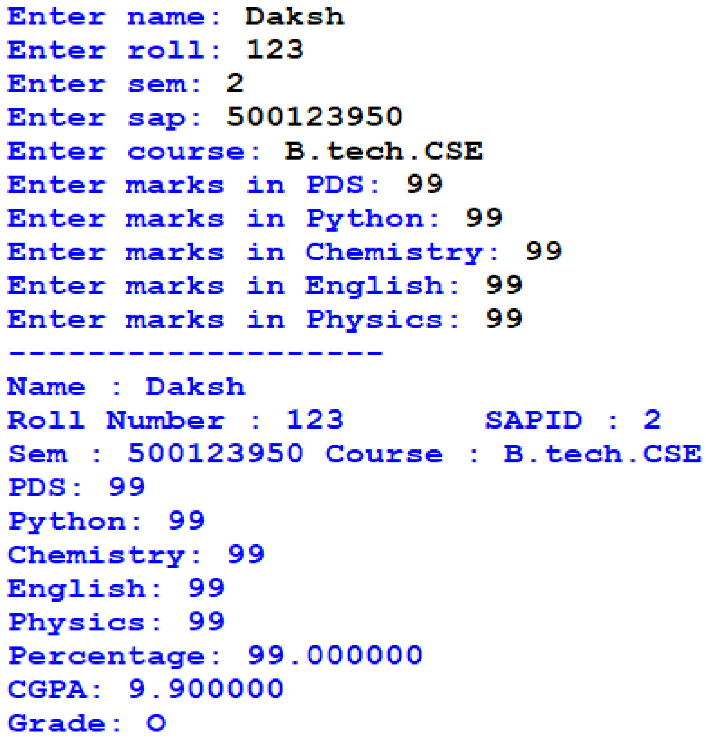
Grade: A

Ans:



```
name=input("Enter name: ")
roll=input("Enter roll: ")
sem=input("Enter sem: ")
sap=input("Enter sap: ")
course=input("Enter course: ")
a=int(input("Enter marks in PDS: "))
b=int(input("Enter marks in Python: "))
c=int(input("Enter marks in Chemistry: "))
d=int(input("Enter marks in English: "))
e=int(input("Enter marks in Physics: "))
percent=(a+b+c+d+e)/5
cgpa=percent/10
if cgpa>=0 and cgpa<=3.4:
        grade='F'
elif cgpa>=3.5 and cgpa<=5:
        grade='C+'
elif cgpa>=5.1 and cgpa<=6:
        grade='B'
elif cgpa>=6.1 and cgpa<=7:
        grade='B+'
elif cgpa>=7.1 and cgpa<=8:
        grade='A'
elif cgpa>=8.1 and cgpa<=9:
        grade='A+'
else:
        grade='O'
print('-------------------')
print('Name : %s\nRoll Number : %s\tSAPID : %s\nSem : %s\tCourse : %s'%(name,roll,sem,sap,course))
print("PDS: %d\nPython: %d\nChemistry: %d\nEnglish: %d\nPhysics: %d"%(a,b,c,d,e))
print('Percentage: %f\nCGPA: %f\nGrade: %s'%(percent,cgpa,grade))
```

Output:

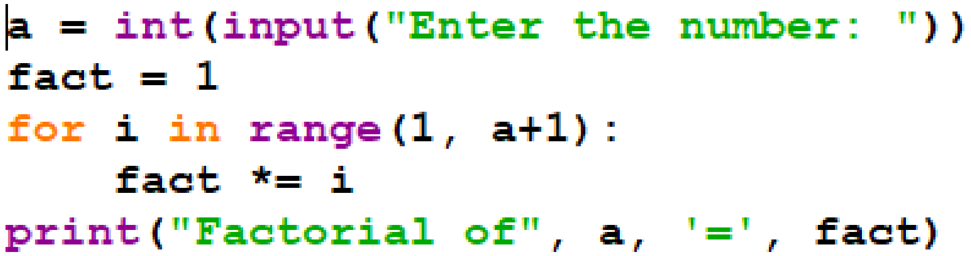


```
Enter name: Daksh
Enter roll: 123
Enter sem: 2
Enter sap: 500123950
Enter course: B.tech.CSE
Enter marks in PDS: 99
Enter marks in Python: 99
Enter marks in Chemistry: 99
Enter marks in English: 99
Enter marks in Physics: 99
--------------------
Name : Daksh
Roll Number : 123       SAPID : 2
Sem : 500123950 Course : B.tech.CSE
PDS: 99
Python: 99
Chemistry: 99
English: 99
Physics: 99
Percentage: 99.000000
CGPA: 9.900000
Grade: O
```

Lab – 4

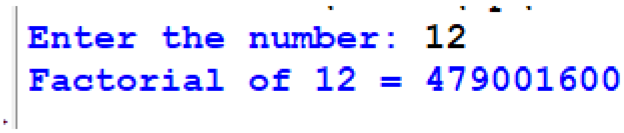
1) Find a factorial of given number.

Ans:



```
a = int(input("Enter the number: "))
fact = 1
for i in range(1, a+1):
    fact *= i
print("Factorial of", a, '=', fact)
```

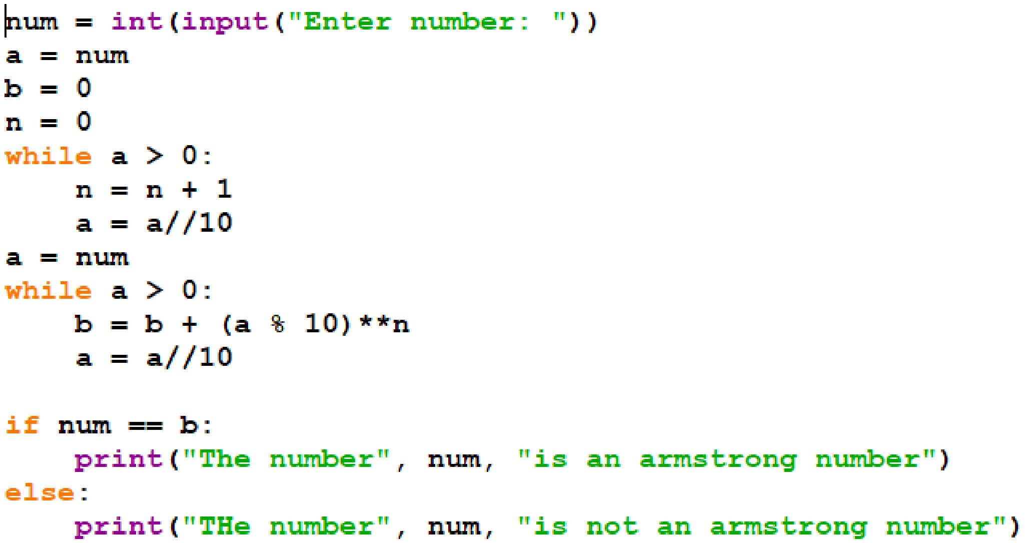
Output:



```
Enter the number: 12
Factorial of 12 = 479001600
```

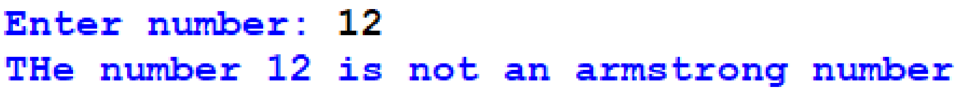
2) Find whether the given number is Armstrong number.

Ans:



```
num = int(input("Enter number: "))
a = num
b = 0
n = 0
while a > 0:
    n = n + 1
    a = a//10
a = num
while a > 0:
    b = b + (a % 10)**n
    a = a//10

if num == b:
    print("The number", num, "is an armstrong number")
else:
    print("THe number", num, "is not an armstrong number")
```

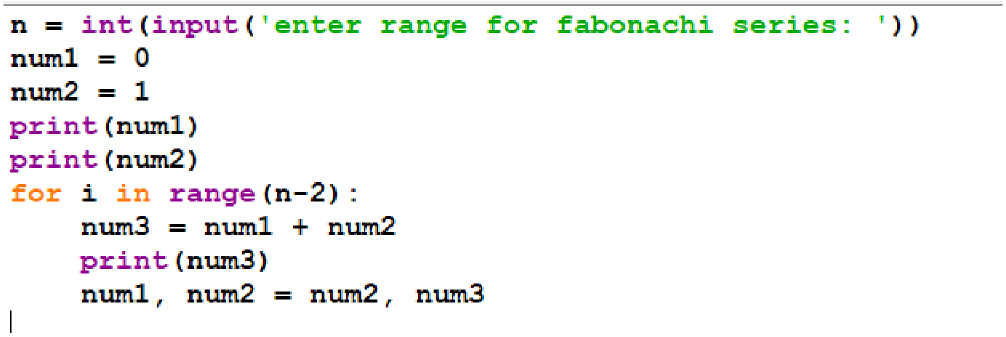
Output:



```
Enter number: 12
THe number 12 is not an armstrong number
```

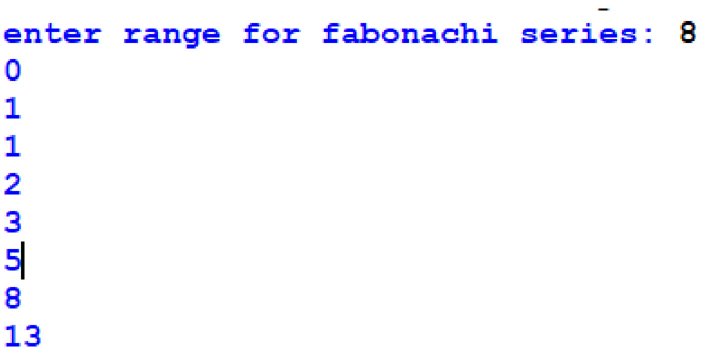
3) Print Fibonacci series up to given term.

Ans:



```
n = int(input('enter range for fabonachi series: '))
num1 = 0
num2 = 1
print(num1)
print(num2)
for i in range(n-2):
    num3 = num1 + num2
    print(num3)
    num1, num2 = num2, num3
```

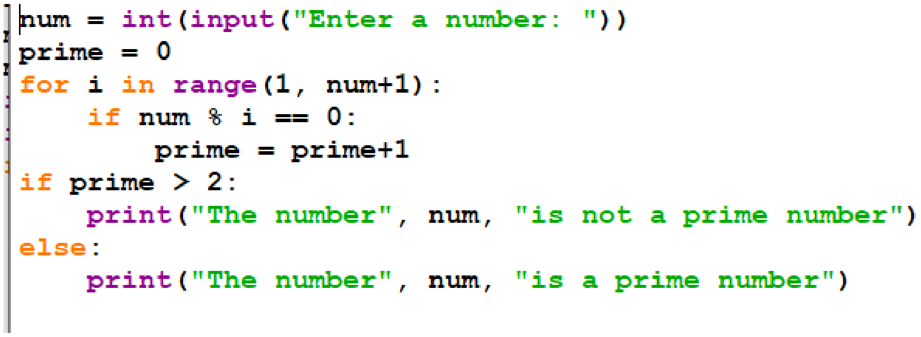
Output:



```
enter range for fabonachi series: 8
0
1
1
2
3
5
8
13
```

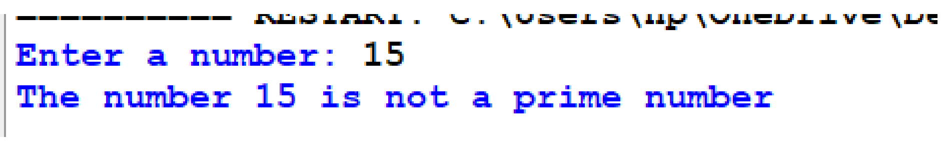
4) Write a program to find if given number is prime number or not.

Ans:



```
num = int(input("Enter a number: "))
prime = 0
for i in range(1, num+1):
    if num % i == 0:
        prime = prime+1
if prime > 2:
    print("The number", num, "is not a prime number")
else:
    print("The number", num, "is a prime number")
```

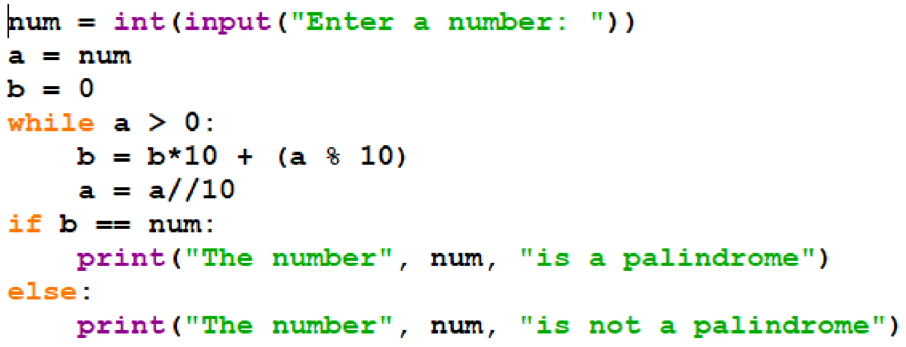
Output:



```
========== RESTART: C:\Users\hp\OneDrive\De
Enter a number: 15
The number 15 is not a prime number
```

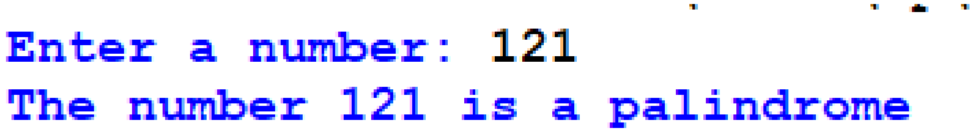
5) Check whether given number is palindrome or not.

Ans:



```
num = int(input("Enter a number: "))
a = num
b = 0
while a > 0:
    b = b*10 + (a % 10)
    a = a//10
if b == num:
    print("The number", num, "is a palindrome")
else:
    print("The number", num, "is not a palindrome")
```

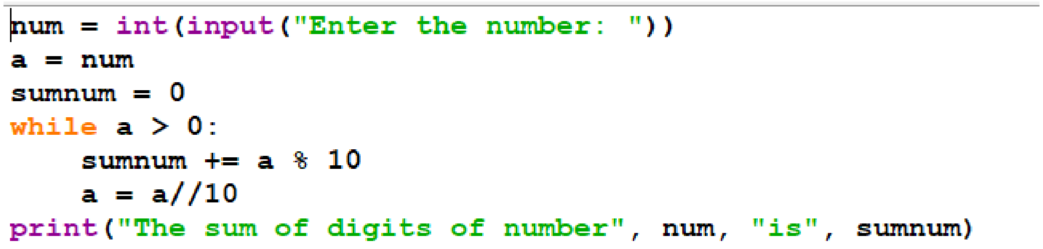
Output:



```
Enter a number: 121
The number 121 is a palindrome
```

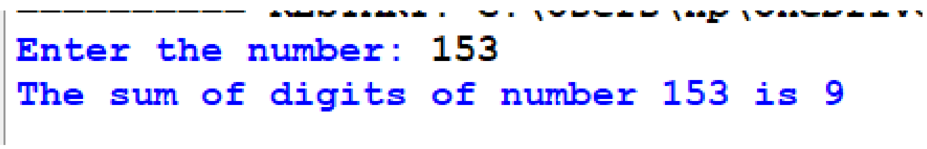
6) Write a program to print sum of digits.

Ans:



```
num = int(input("Enter the number: "))
a = num
sumnum = 0
while a > 0:
    sumnum += a % 10
    a = a//10
print("The sum of digits of number", num, "is", sumnum)
```

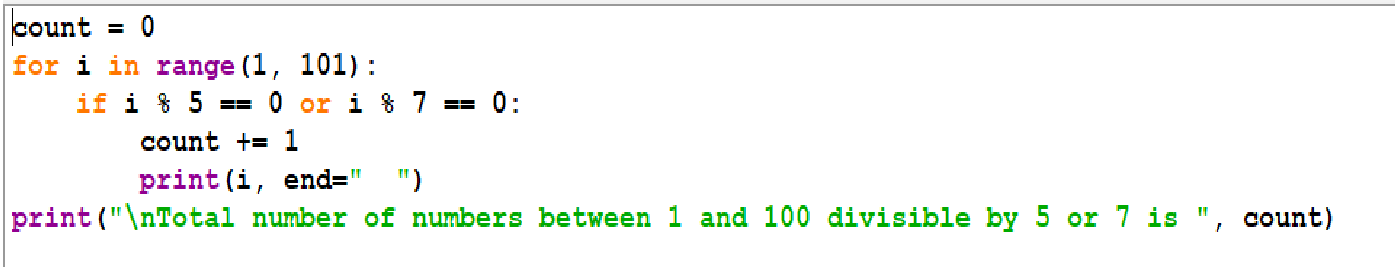
Output:



```
Enter the number: 153
The sum of digits of number 153 is 9
```

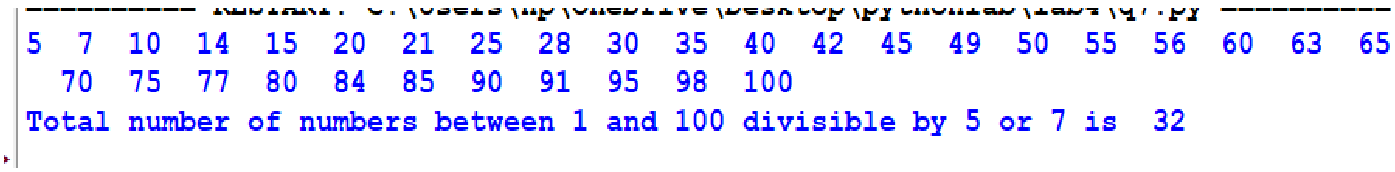
7) Count and print all numbers divisible by 5 or 7 between 1 to 100.

Ans:



```
count = 0
for i in range(1, 101):
    if i % 5 == 0 or i % 7 == 0:
        count += 1
        print(i, end="  ")
print("\nTotal number of numbers between 1 and 100 divisible by 5 or 7 is ", count)
```

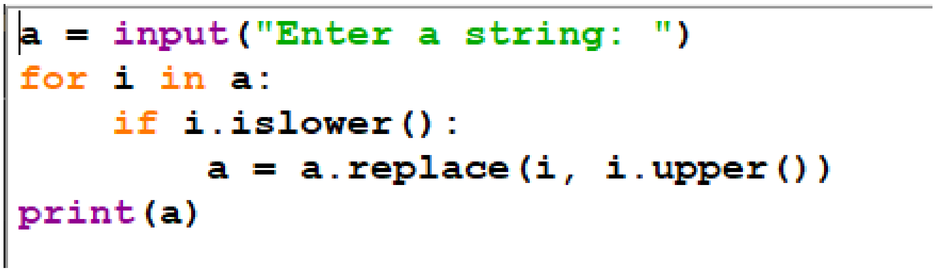
Output:



```
========== RESTART: C:\Users\hp\OneDrive\Desktop\pythonlab\lab4\q7.py ==========
5  7  10  14  15  20  21  25  28  30  35  40  42  45  49  50  55  56  60  63  65
  70  75  77  80  84  85  90  91  95  98  100
Total number of numbers between 1 and 100 divisible by 5 or 7 is  32
```

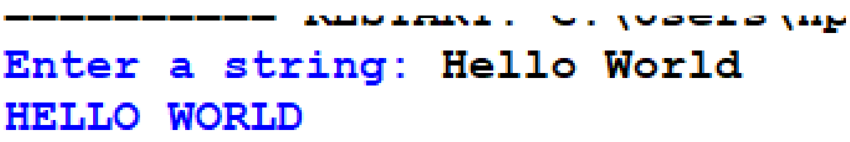
8) Convert all lower cases to upper case in a string.

Ans:



```
a = input("Enter a string: ")
for i in a:
    if i.islower():
        a = a.replace(i, i.upper())
print(a)
```

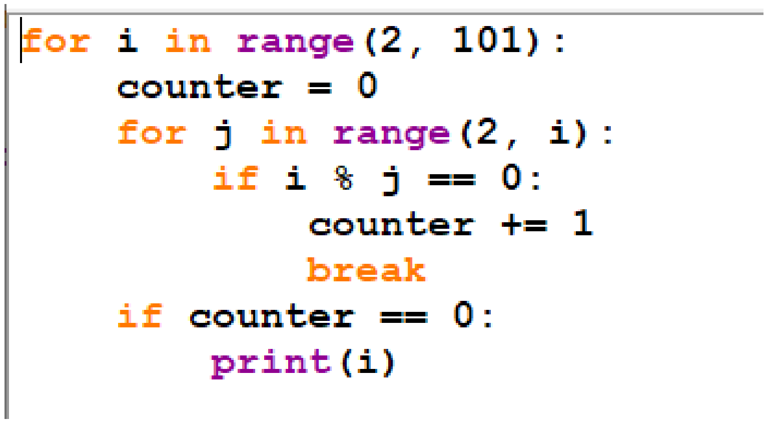
Output:



```
=========== RESTART: C:\Users\hp
Enter a string: Hello World
HELLO WORLD
```

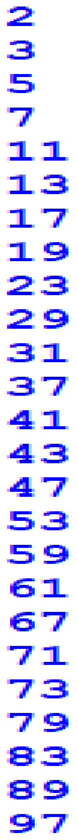
9) Print all prime numbers between 1 and 100.

Ans:



```
for i in range(2, 101):
    counter = 0
    for j in range(2, i):
        if i % j == 0:
            counter += 1
            break
    if counter == 0:
        print(i)
```

Output:\



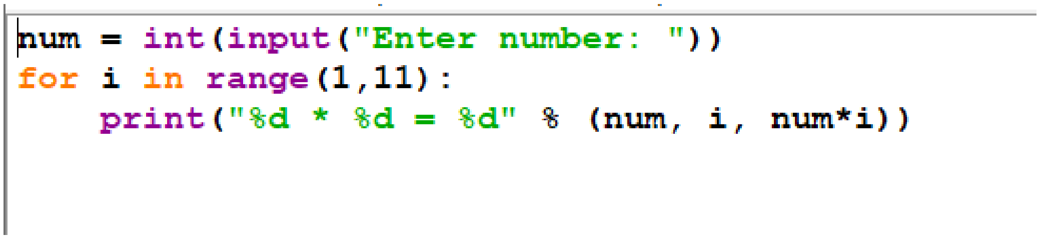
```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

10) Print the table for a given number:

5 * 1 = 5

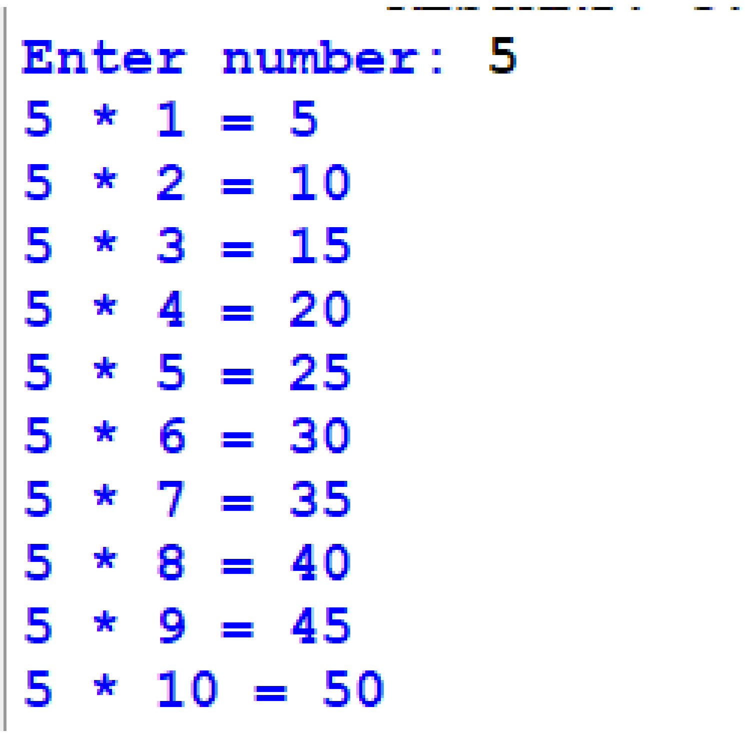
5 * 2 = 10………..

Ans:



```
num = int(input("Enter number: "))
for i in range(1,11):
    print("%d * %d = %d" % (num, i, num*i))
```

Output:

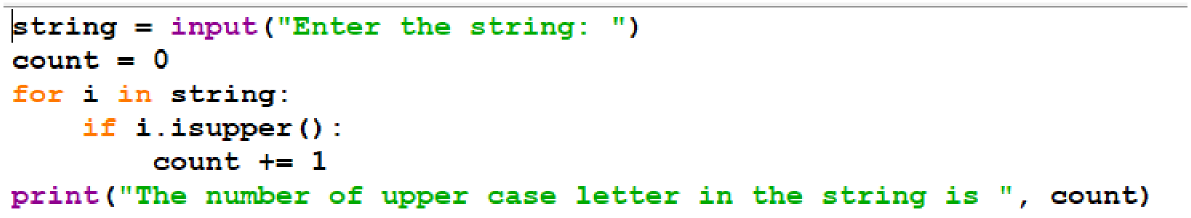


```
Enter number: 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Lab – 5

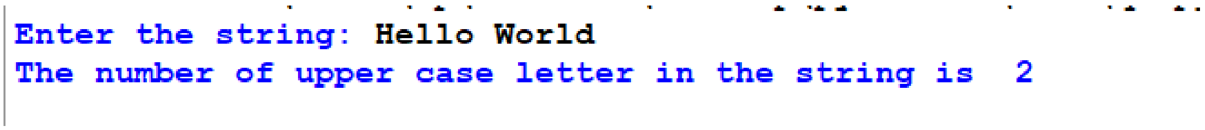
1) Write a program to count and display the number of capital letters in a given string.

Ans:



```python
string = input("Enter the string: ")
count = 0
for i in string:
    if i.isupper():
        count += 1
print("The number of upper case letter in the string is ", count)
```

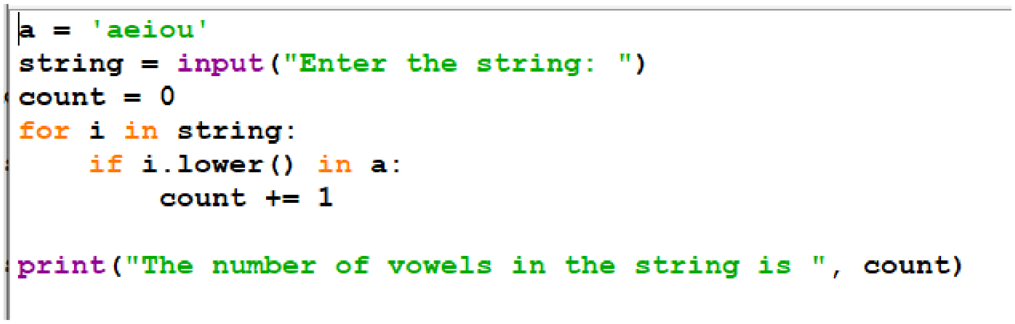
Output:



```
Enter the string: Hello World
The number of upper case letter in the string is  2
```

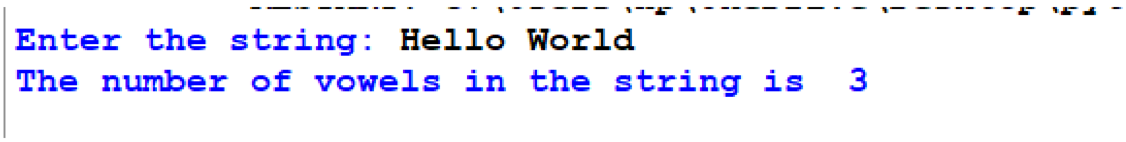
2.)Count total number of vowels in a given string.

Ans:



```python
a = 'aeiou'
string = input("Enter the string: ")
count = 0
for i in string:
    if i.lower() in a:
        count += 1

print("The number of vowels in the string is ", count)
```

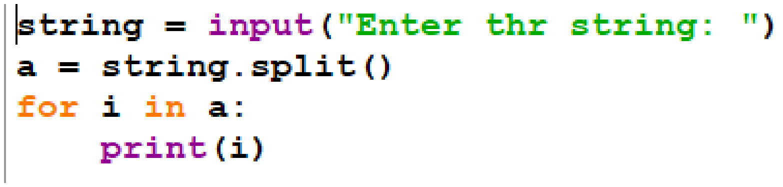
Output:



```
Enter the string: Hello World
The number of vowels in the string is  3
```

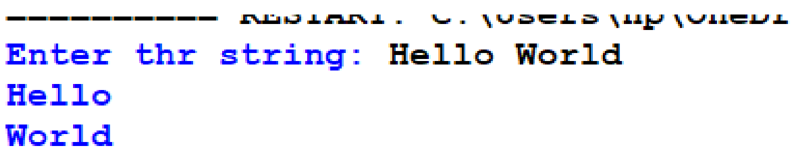
3) Input a sentence and print words in separate lines.

Ans:



```
string = input("Enter thr string: ")
a = string.split()
for i in a:
    print(i)
```

Output:



```
========== RESTART: C:\Users\hp\OneDri
Enter thr string: Hello World
Hello
World
```

4) WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given

string. String traversal will take place from left to right, not from right to left.

Sample Input

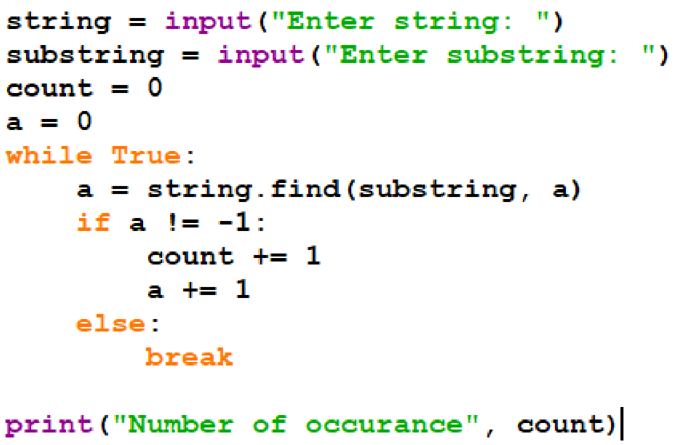
ABCDCDC

CDC

Sample Output

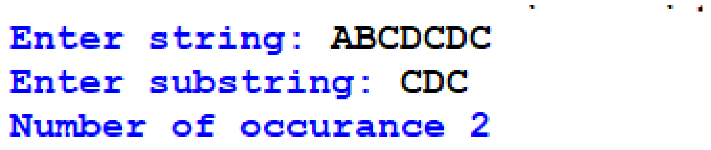
2

Ans:



```
string = input("Enter string: ")
substring = input("Enter substring: ")
count = 0
a = 0
while True:
    a = string.find(substring, a)
    if a != -1:
        count += 1
        a += 1
    else:
        break

print("Number of occurance", count)
```

Output:



```
Enter string: ABCDCDC
Enter substring: CDC
Number of occurance 2
```

5) Given a string containing both upper and lower case alphabets. Write a Python program to count the number of

occurrences of each alphabet (case insensitive) and display the same.

Sample Input

ABaBCbGc

Sample Output

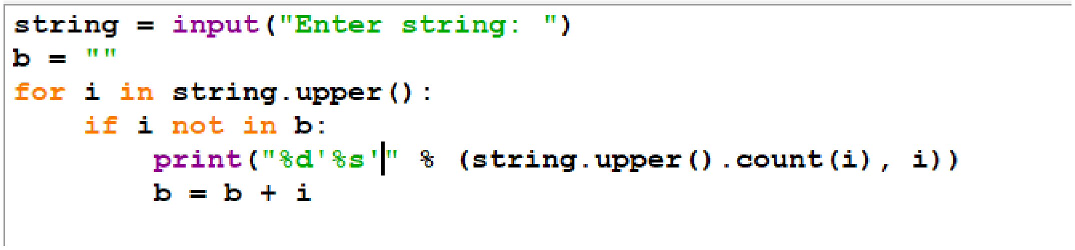
2A

3B

2C

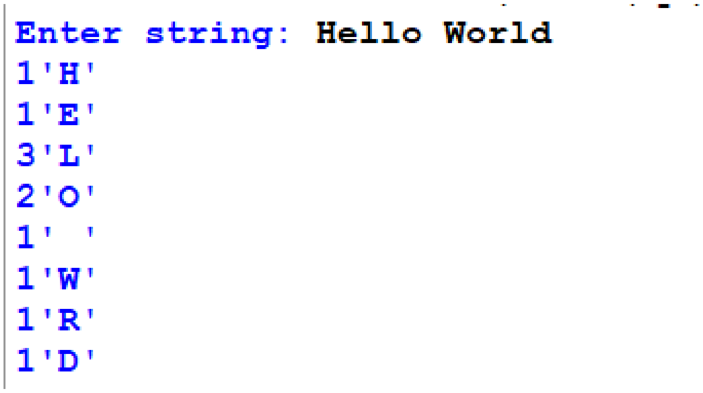
1G

Ans:



```python
string = input("Enter string: ")
b = ""
for i in string.upper():
    if i not in b:
        print("%d'%s'" % (string.upper().count(i), i))
        b = b + i
```

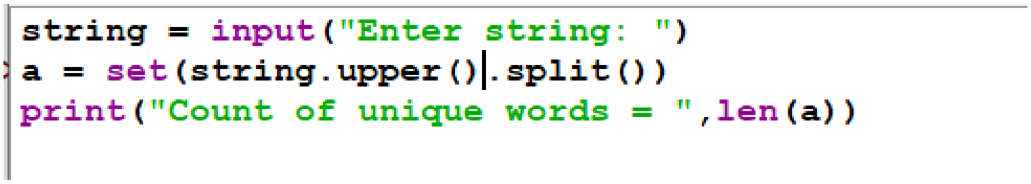
Output:



```
Enter string: Hello World
1'H'
1'E'
3'L'
2'O'
1' '
1'W'
1'R'
1'D'
```

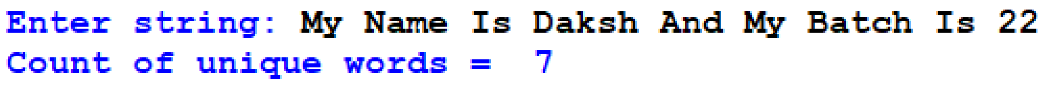
6) Program to count number of unique words in a given sentence using sets.

Ans:



```
string = input("Enter string: ")
a = set(string.upper().split())
print("Count of unique words = ",len(a))
```

Output:



```
Enter string: My Name Is Daksh And My Batch Is 22
Count of unique words =  7
```

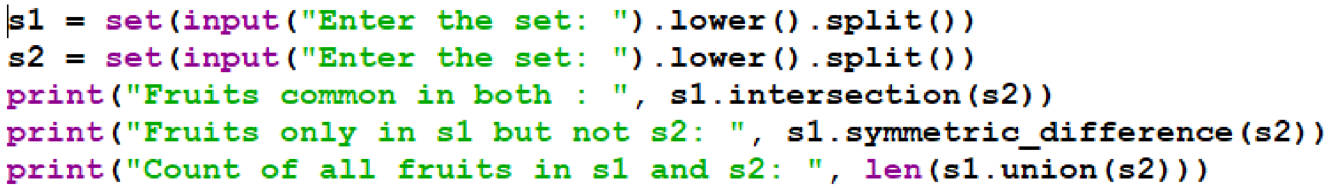
7) Create 2 sets s1 and s2 of n fruits each by taking input from user and find:

a. Fruits which are in both sets s1 and s2

b. Fruits only in s1 but not in s2

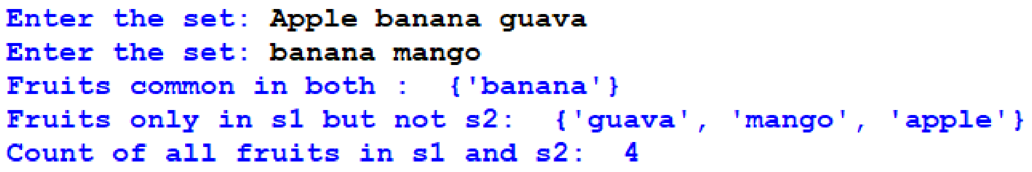
c. Count of all fruits from s1 and s2

Ans:



```
s1 = set(input("Enter the set: ").lower().split())
s2 = set(input("Enter the set: ").lower().split())
print("Fruits common in both : ", s1.intersection(s2))
print("Fruits only in s1 but not s2: ", s1.symmetric_difference(s2))
print("Count of all fruits in s1 and s2: ", len(s1.union(s2)))
```

Output:



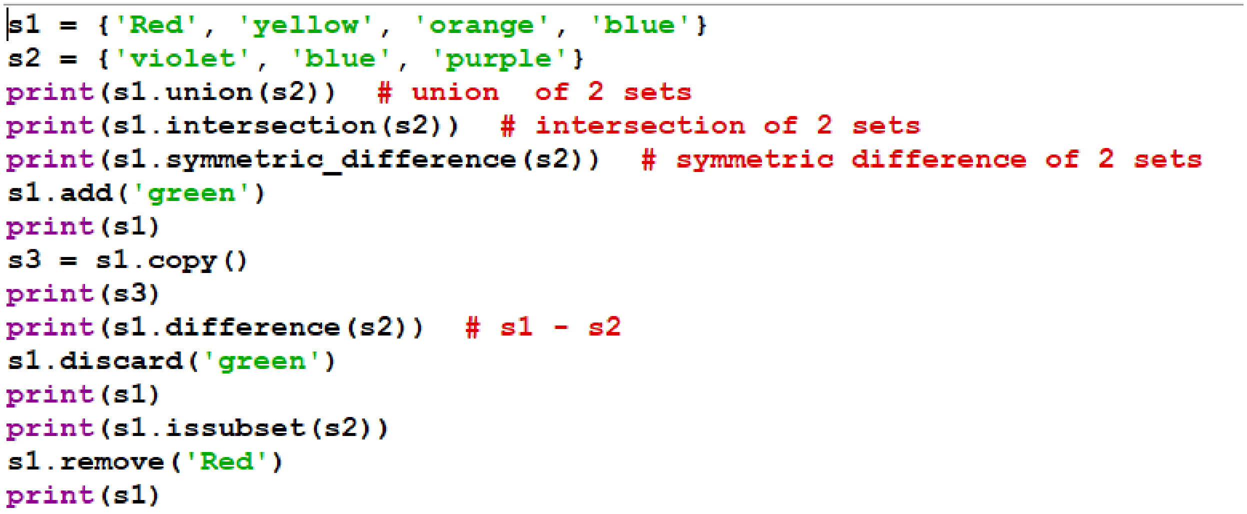
```
Enter the set: Apple banana guava
Enter the set: banana mango
Fruits common in both :  {'banana'}
Fruits only in s1 but not s2:  {'guava', 'mango', 'apple'}
Count of all fruits in s1 and s2:  4
```

8) Take two sets and apply various set operations on them :

S1 = {Red ,yellow, orange , blue }

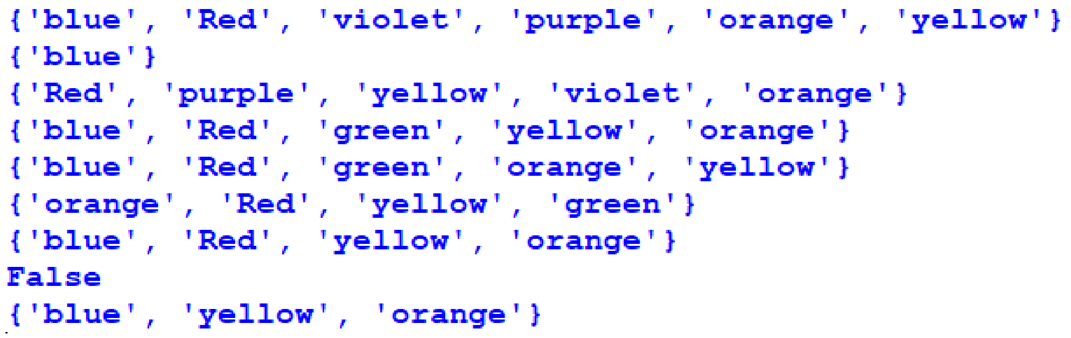
S2 = {violet, blue , purple}

Ans:



```
s1 = {'Red', 'yellow', 'orange', 'blue'}
s2 = {'violet', 'blue', 'purple'}
print(s1.union(s2))  # union  of 2 sets
print(s1.intersection(s2))  # intersection of 2 sets
print(s1.symmetric_difference(s2))  # symmetric difference of 2 sets
s1.add('green')
print(s1)
s3 = s1.copy()
print(s3)
print(s1.difference(s2))  # s1 - s2
s1.discard('green')
print(s1)
print(s1.issubset(s2))
s1.remove('Red')
print(s1)
```

Output:

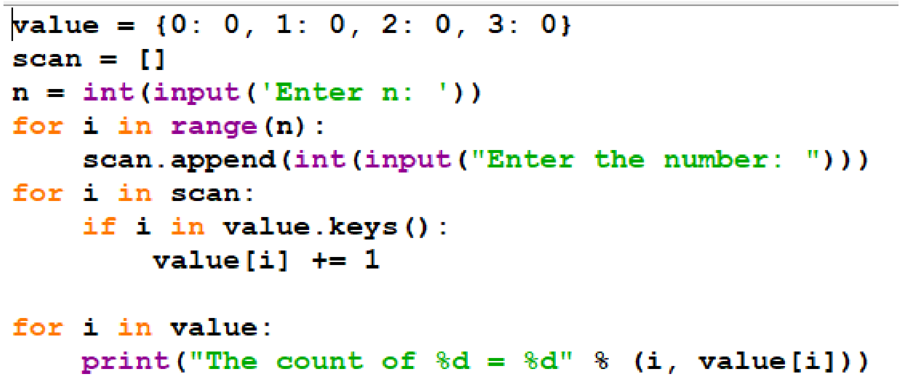


```
{'blue', 'Red', 'violet', 'purple', 'orange', 'yellow'}
{'blue'}
{'Red', 'purple', 'yellow', 'violet', 'orange'}
{'blue', 'Red', 'green', 'yellow', 'orange'}
{'blue', 'Red', 'green', 'orange', 'yellow'}
{'orange', 'Red', 'yellow', 'green'}
{'blue', 'Red', 'yellow', 'orange'}
False
{'blue', 'yellow', 'orange'}
```

Lab – 6

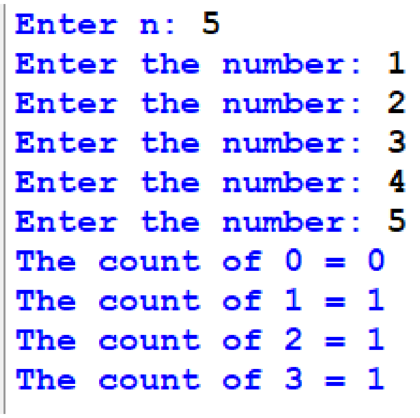
1) Scan n values in range 0-3 and print the number of times each value has occurred.

Ans:



```python
value = {0: 0, 1: 0, 2: 0, 3: 0}
scan = []
n = int(input('Enter n: '))
for i in range(n):
    scan.append(int(input("Enter the number: ")))
for i in scan:
    if i in value.keys():
        value[i] += 1

for i in value:
    print("The count of %d = %d" % (i, value[i]))
```

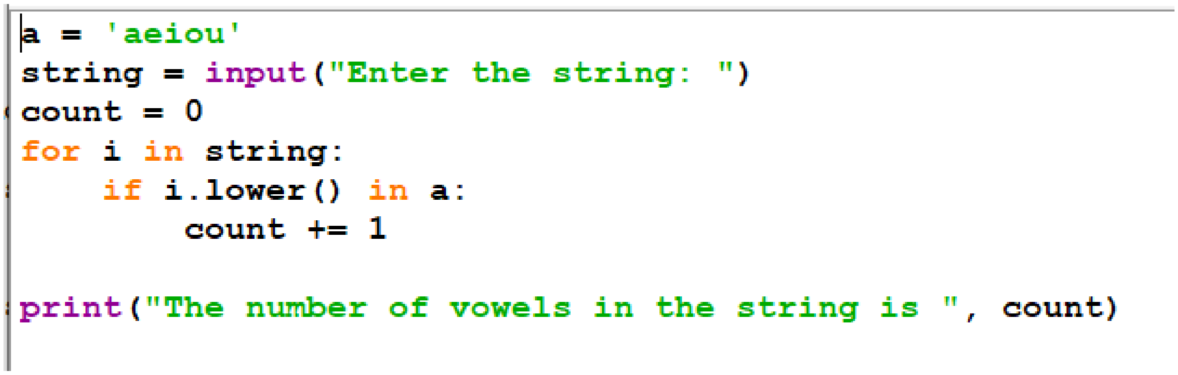
Output:



```
Enter n: 5
Enter the number: 1
Enter the number: 2
Enter the number: 3
Enter the number: 4
Enter the number: 5
The count of 0 = 0
The count of 1 = 1
The count of 2 = 1
The count of 3 = 1
```

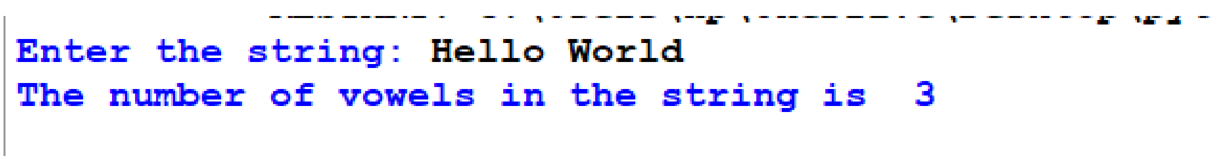
2) Create a tuple to store n numeric values and find average of all values.

Ans:



```
a = 'aeiou'
string = input("Enter the string: ")
count = 0
for i in string:
    if i.lower() in a:
        count += 1

print("The number of vowels in the string is ", count)
```

Output:



```
Enter the string: Hello World
The number of vowels in the string is  3
```

3) WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.

Sample Input

N = 5

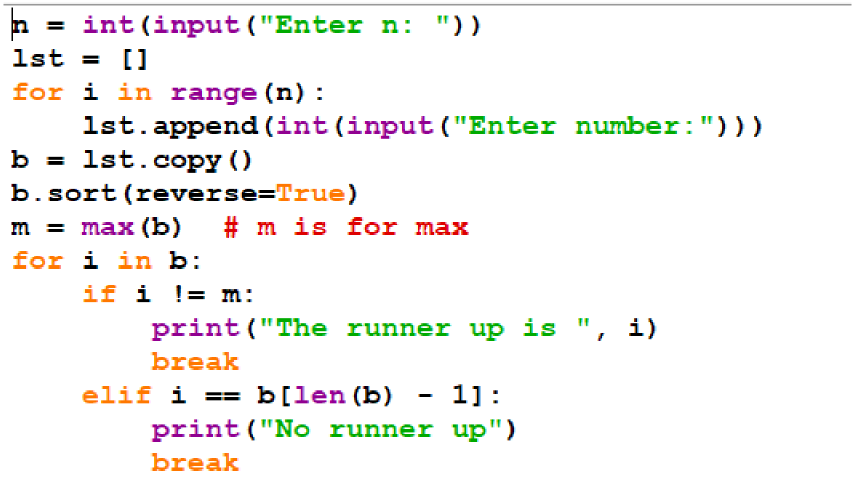
Scores= 2 3 6 6 5

Sample output

5

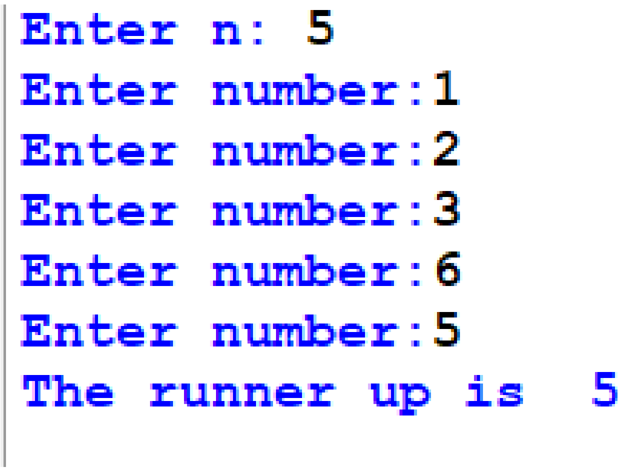
Note: Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

Ans:



```
n = int(input("Enter n: "))
lst = []
for i in range(n):
    lst.append(int(input("Enter number:")))
b = lst.copy()
b.sort(reverse=True)
m = max(b)   # m is for max
for i in b:
    if i != m:
        print("The runner up is ", i)
        break
    elif i == b[len(b) - 1]:
        print("No runner up")
        break
```

Output:



```
Enter n:  5
Enter number:1
Enter number:2
Enter number:3
Enter number:6
Enter number:5
The runner up is  5
```

4) Create a dictionary of n persons where key is name and value is city.

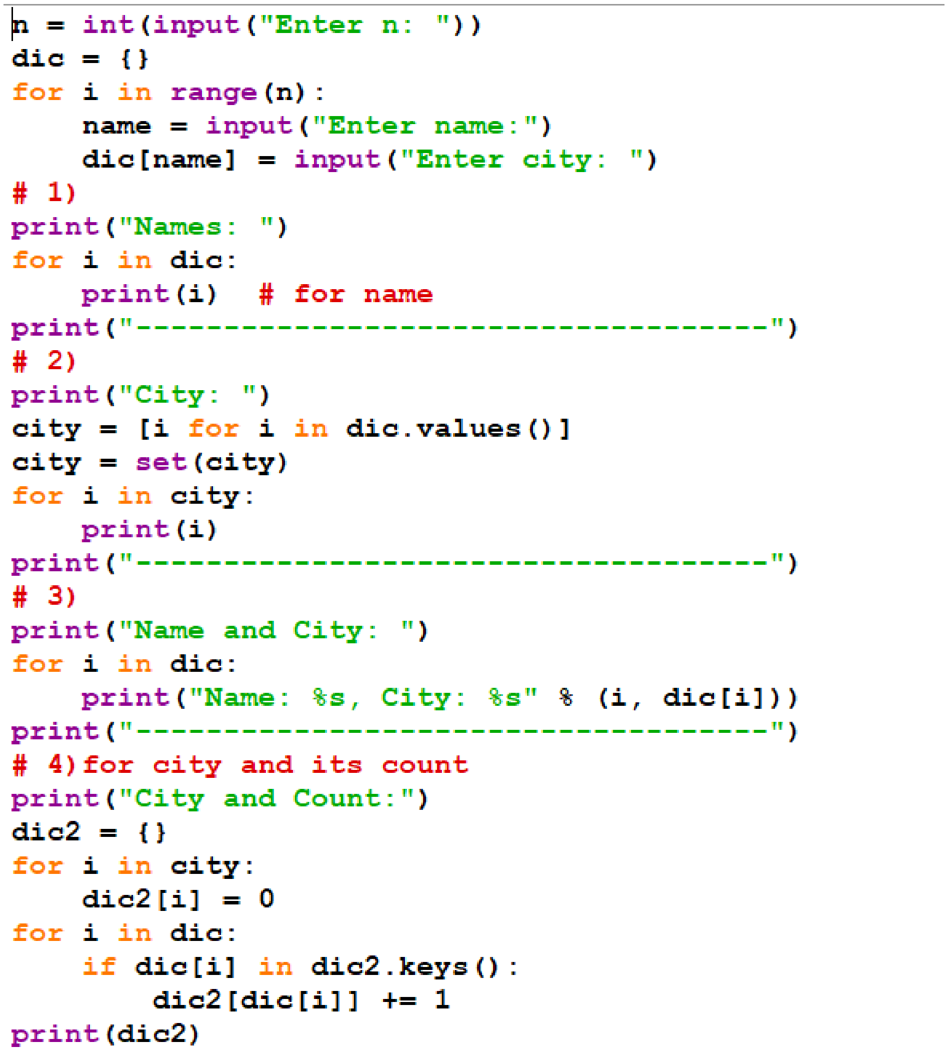
a. Display all names

b. Display all city names

c. Display student name and city of all students.

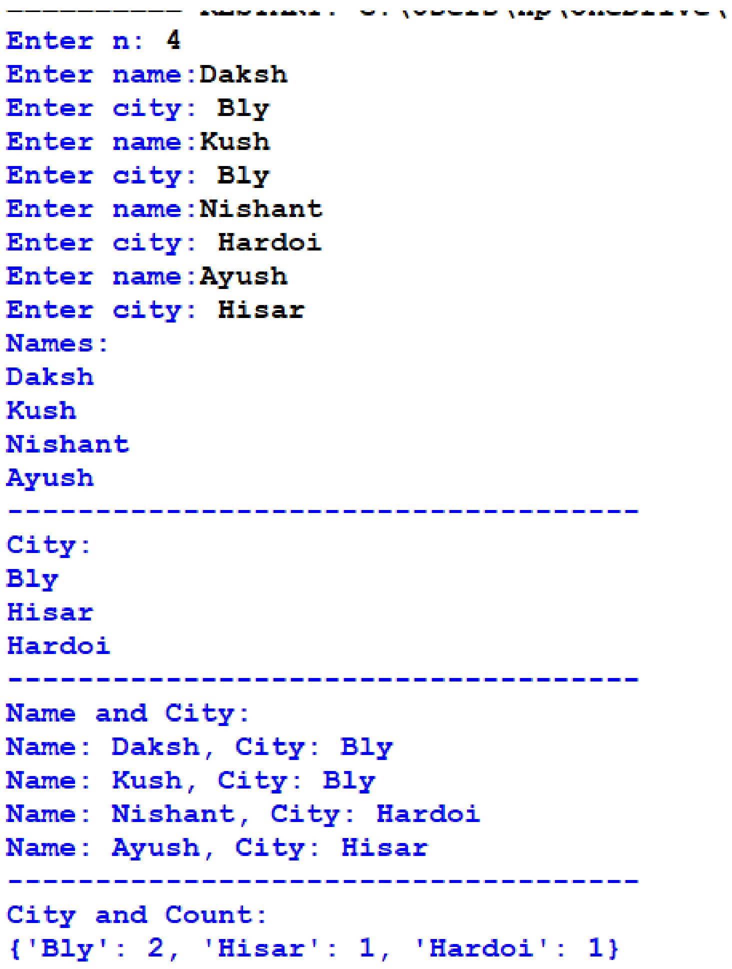
d. Count number of students in each city.

Ans:



```python
n = int(input("Enter n: "))
dic = {}
for i in range(n):
    name = input("Enter name:")
    dic[name] = input("Enter city: ")
# 1)
print("Names: ")
for i in dic:
    print(i)  # for name
print("------------------------------------")
# 2)
print("City: ")
city = [i for i in dic.values()]
city = set(city)
for i in city:
    print(i)
print("------------------------------------")
# 3)
print("Name and City: ")
for i in dic:
    print("Name: %s, City: %s" % (i, dic[i]))
print("------------------------------------")
# 4)for city and its count
print("City and Count:")
dic2 = {}
for i in city:
    dic2[i] = 0
for i in dic:
    if dic[i] in dic2.keys():
        dic2[dic[i]] += 1
print(dic2)
```

Output:



```
Enter n: 4
Enter name:Daksh
Enter city: Bly
Enter name:Kush
Enter city: Bly
Enter name:Nishant
Enter city: Hardoi
Enter name:Ayush
Enter city: Hisar
Names:
Daksh
Kush
Nishant
Ayush
------------------------------------
City:
Bly
Hisar
Hardoi
------------------------------------
Name and City:
Name: Daksh, City: Bly
Name: Kush, City: Bly
Name: Nishant, City: Hardoi
Name: Ayush, City: Hisar
------------------------------------
City and Count:
{'Bly': 2, 'Hisar': 1, 'Hardoi': 1}
```

5) Store details of n movies in a dictionary by taking input from the user. Each movie must store details like name, year,

director name, production cost, collection made (earning) & perform the following :-

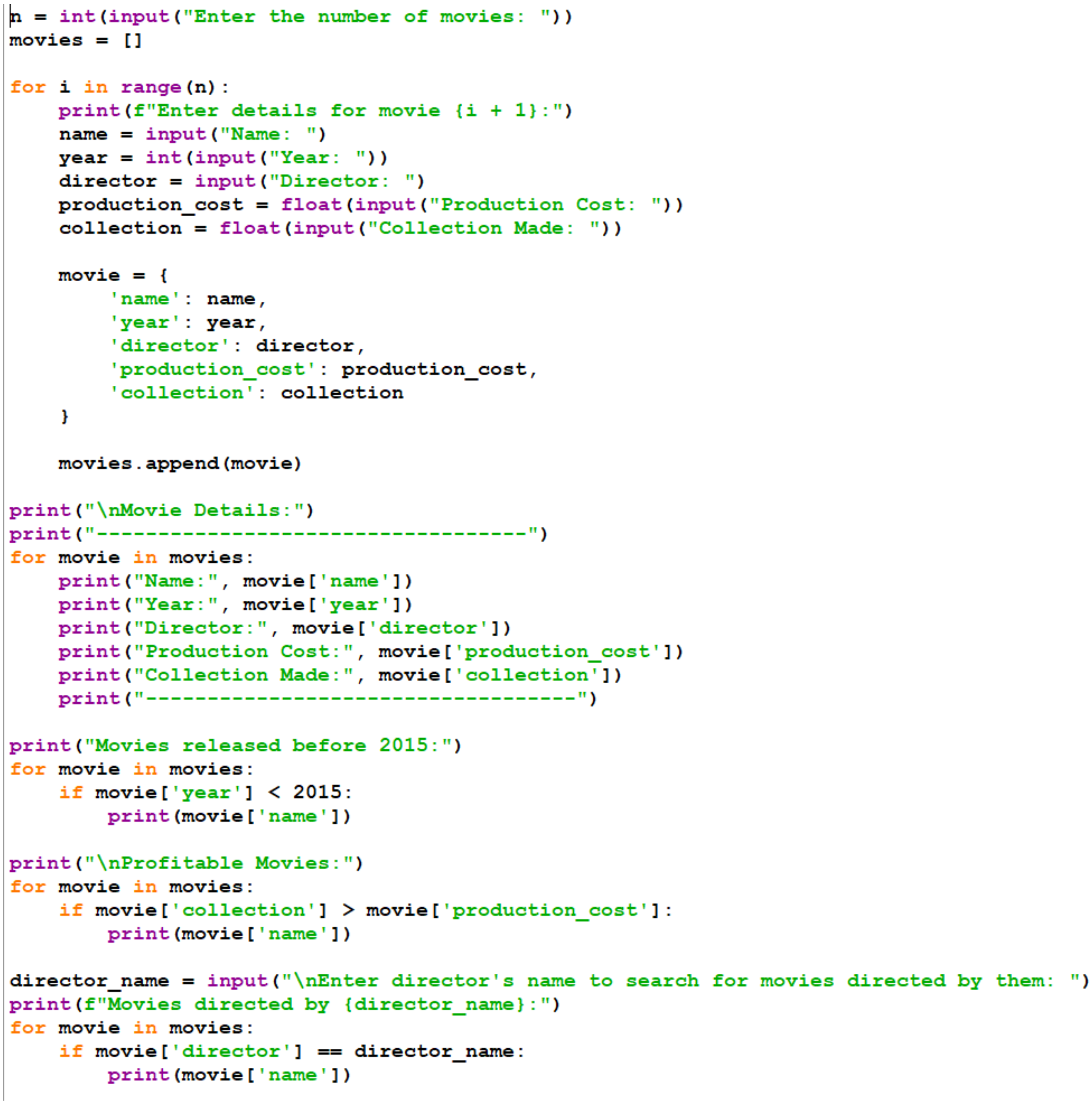
a. print all movie details

b. display name of movies released before 2015

c. print movies that made a profit.

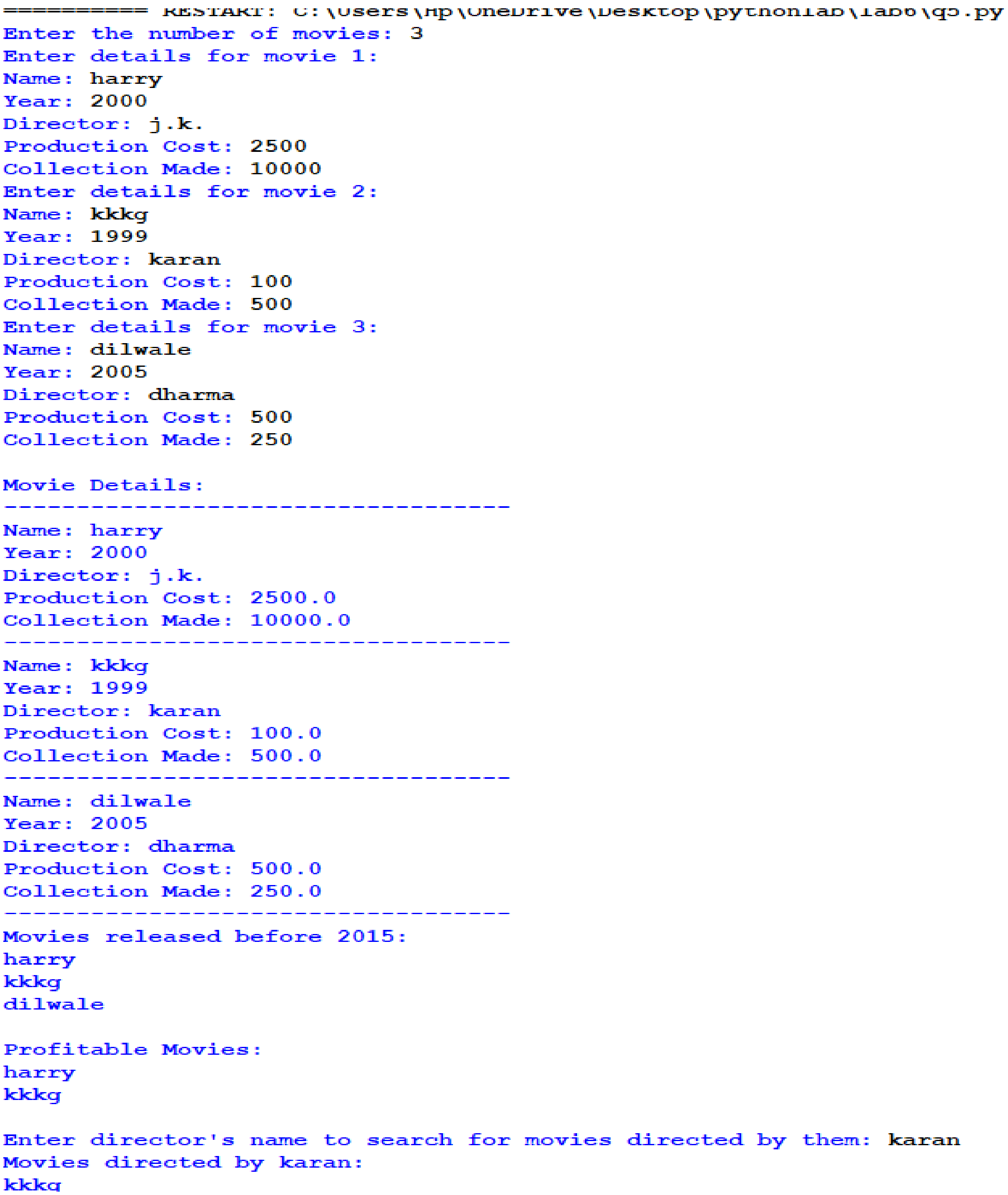
d. print movies directed by a particular director

Ans:



```python
n = int(input("Enter the number of movies: "))
movies = []

for i in range(n):
    print(f"Enter details for movie {i + 1}:")
    name = input("Name: ")
    year = int(input("Year: "))
    director = input("Director: ")
    production_cost = float(input("Production Cost: "))
    collection = float(input("Collection Made: "))

    movie = {
        'name': name,
        'year': year,
        'director': director,
        'production_cost': production_cost,
        'collection': collection
    }

    movies.append(movie)

print("\nMovie Details:")
print("-----------------------------------")
for movie in movies:
    print("Name:", movie['name'])
    print("Year:", movie['year'])
    print("Director:", movie['director'])
    print("Production Cost:", movie['production_cost'])
    print("Collection Made:", movie['collection'])
    print("-----------------------------------")

print("Movies released before 2015:")
for movie in movies:
    if movie['year'] < 2015:
        print(movie['name'])

print("\nProfitable Movies:")
for movie in movies:
    if movie['collection'] > movie['production_cost']:
        print(movie['name'])

director_name = input("\nEnter director's name to search for movies directed by them: ")
print(f"Movies directed by {director_name}:")
for movie in movies:
    if movie['director'] == director_name:
        print(movie['name'])
```

Output:

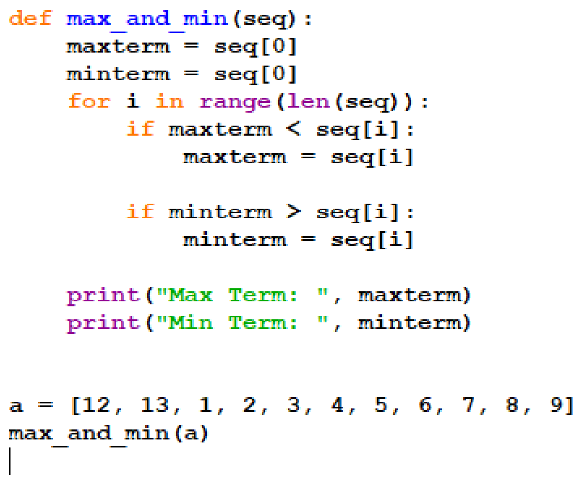


```
========== RESTART: C:\Users\hp\OneDrive\Desktop\pythonlab\lab6\q5.py
Enter the number of movies: 3
Enter details for movie 1:
Name: harry
Year: 2000
Director: j.k.
Production Cost: 2500
Collection Made: 10000
Enter details for movie 2:
Name: kkkg
Year: 1999
Director: karan
Production Cost: 100
Collection Made: 500
Enter details for movie 3:
Name: dilwale
Year: 2005
Director: dharma
Production Cost: 500
Collection Made: 250

Movie Details:
-----------------------------------
Name: harry
Year: 2000
Director: j.k.
Production Cost: 2500.0
Collection Made: 10000.0
-----------------------------------
Name: kkkg
Year: 1999
Director: karan
Production Cost: 100.0
Collection Made: 500.0
-----------------------------------
Name: dilwale
Year: 2005
Director: dharma
Production Cost: 500.0
Collection Made: 250.0
-----------------------------------
Movies released before 2015:
harry
kkkg
dilwale

Profitable Movies:
harry
kkkg

Enter director's name to search for movies directed by them: karan
Movies directed by karan:
kkkg
```

Lab – 7

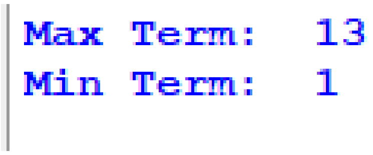
1) Write a Python function to find the maximum and minimum numbers from a sequence of numbers. (Note: Do not use built-in functions.)

Ans:



```
def max_and_min(seq):
    maxterm = seq[0]
    minterm = seq[0]
    for i in range(len(seq)):
        if maxterm < seq[i]:
            maxterm = seq[i]

        if minterm > seq[i]:
            minterm = seq[i]

    print("Max Term: ", maxterm)
    print("Min Term: ", minterm)


a = [12, 13, 1, 2, 3, 4, 5, 6, 7, 8, 9]
max_and_min(a)
```

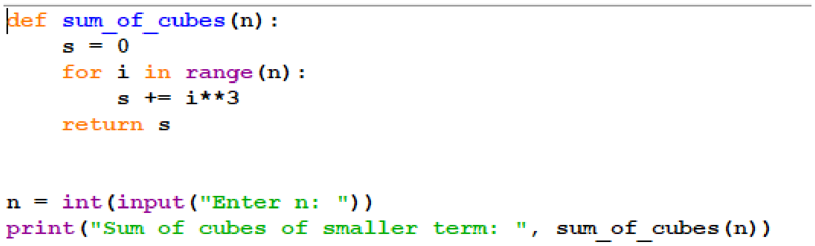
Output:



```
Max Term:  13
Min Term:  1
```

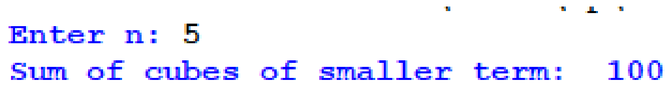
2) Write a Python function that takes a positive integer and returns the sum of the cube of all the positive integers smaller than the specified number.

Ans:



```
def sum_of_cubes(n):
    s = 0
    for i in range(n):
        s += i**3
    return s


n = int(input("Enter n: "))
print("Sum of cubes of smaller term: ", sum_of_cubes(n))
```

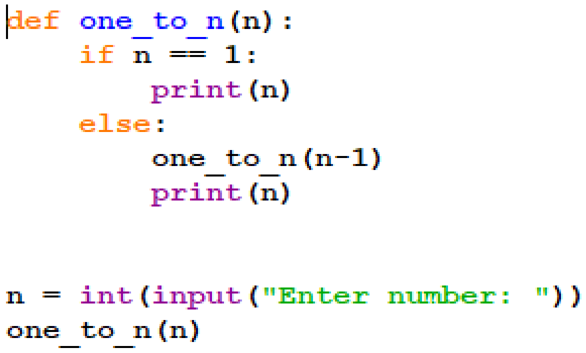
Output:



```
Enter n: 5
Sum of cubes of smaller term:  100
```

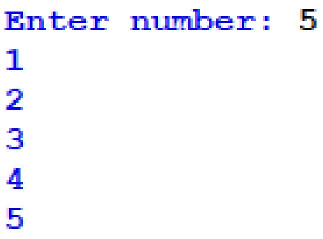
3) Write a Python function to print 1 to n using recursion. (Note: Do not use loop)

Ans:



```
def one_to_n(n):
    if n == 1:
        print(n)
    else:
        one_to_n(n-1)
        print(n)


n = int(input("Enter number: "))
one_to_n(n)
```

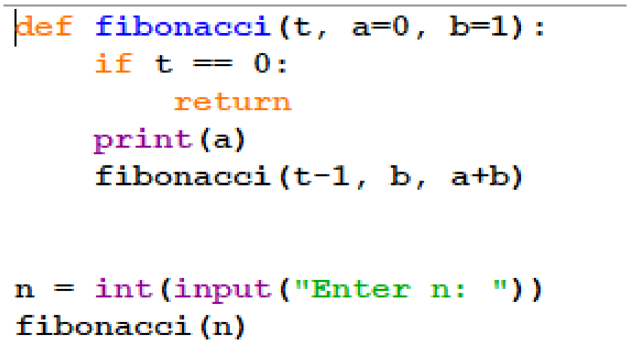
Output:



```
Enter number: 5
1
2
3
4
5
```

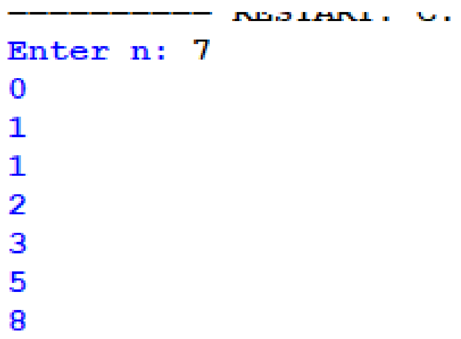
4) Write a recursive function to print Fibonacci series upto n terms.

Ans:



```
def fibonacci(t, a=0, b=1):
    if t == 0:
        return
    print(a)
    fibonacci(t-1, b, a+b)


n = int(input("Enter n: "))
fibonacci(n)
```

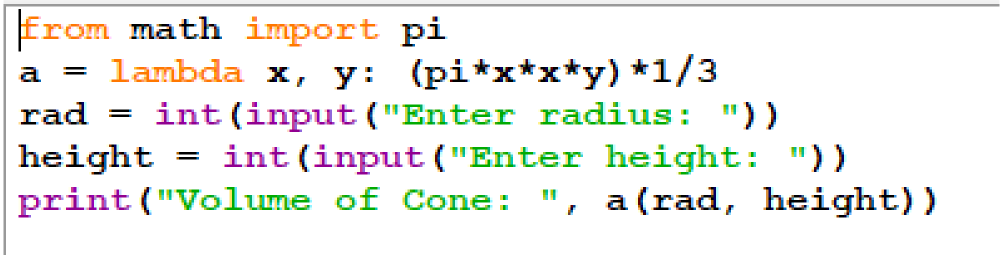
Output:



```
=========== RESTART: C:
Enter n: 7
0
1
1
2
3
5
8
```

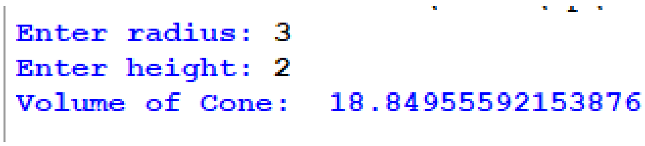
5) Write a lambda function to find volume of cone.

Ans:



```
from math import pi
a = lambda x, y: (pi*x*x*y)*1/3
rad = int(input("Enter radius: "))
height = int(input("Enter height: "))
print("Volume of Cone: ", a(rad, height))
```

Output:



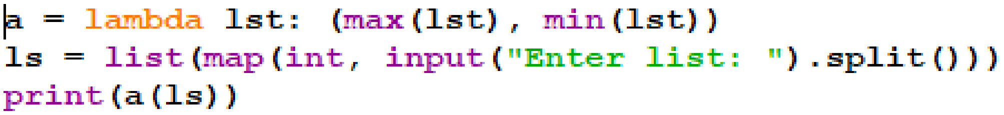
```
Enter radius: 3
Enter height: 2
Volume of Cone:  18.84955592153876
```

6) Write a lambda function which gives tuple of max and min from a list.

Sample input: [10, 6, 8, 90, 12, 56]

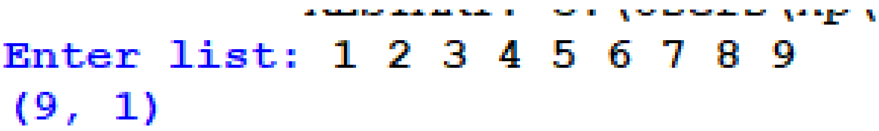
Sample output: (90,6)

Ans:



```
a = lambda lst: (max(lst), min(lst))
ls = list(map(int, input("Enter list: ").split()))
print(a(ls))
```

Output:



```
Enter list: 1 2 3 4 5 6 7 8 9
(9, 1)
```

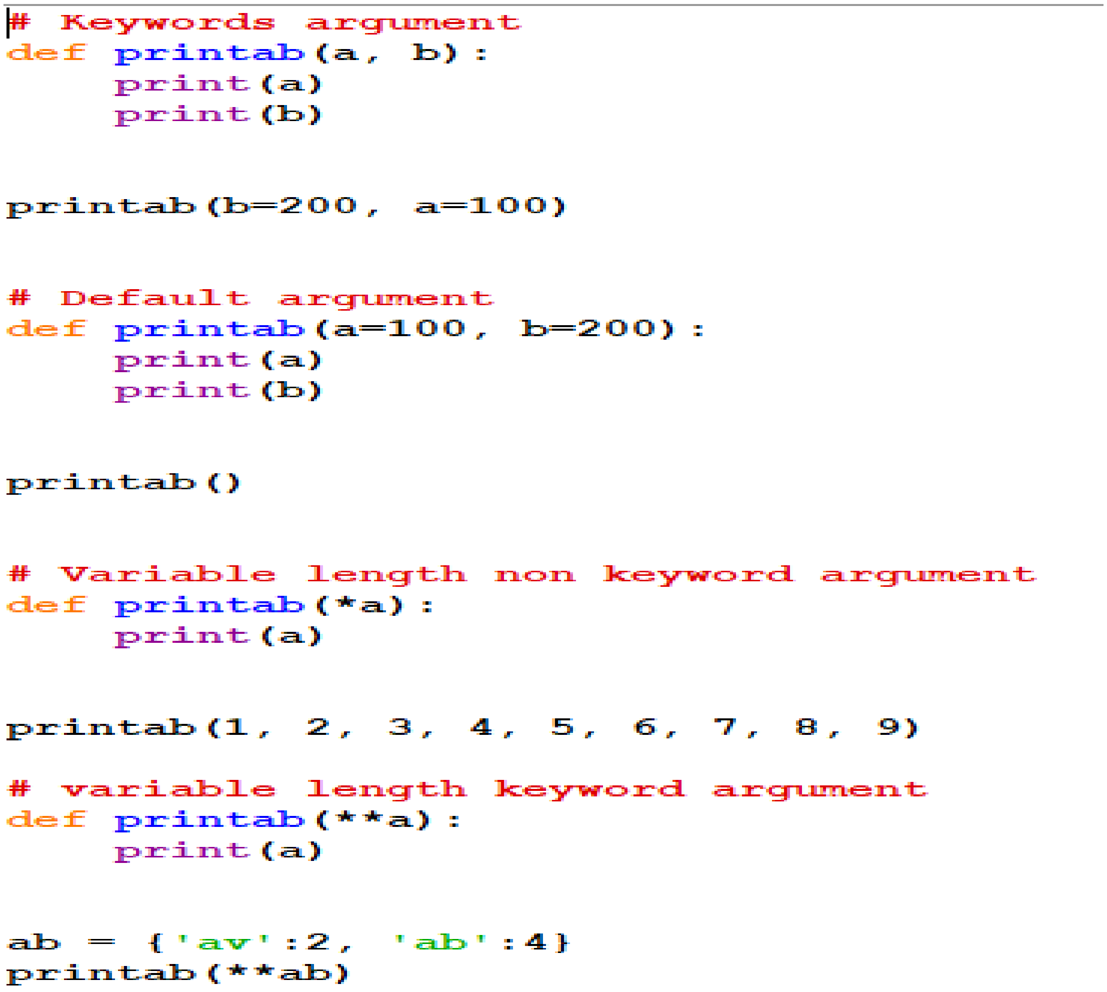
7) Write functions to explain mentioned concepts:

a. Keyword argument

b. Default argument

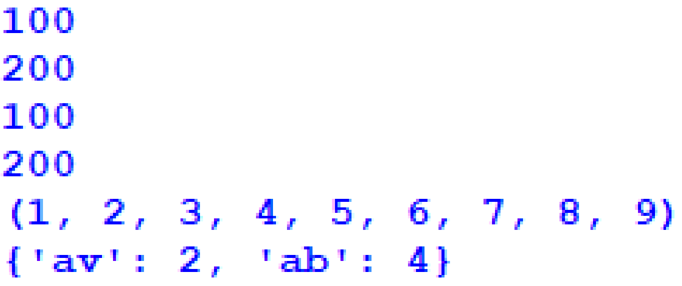
c. Variable length argument

Ans:



```
# Keywords argument
def printab(a, b):
    print(a)
    print(b)


printab(b=200, a=100)


# Default argument
def printab(a=100, b=200):
    print(a)
    print(b)


printab()


# Variable length non keyword argument
def printab(*a):
    print(a)


printab(1, 2, 3, 4, 5, 6, 7, 8, 9)

# variable length keyword argument
def printab(**a):
    print(a)


ab = {'av':2, 'ab':4}
printab(**ab)
```

Output:



```
100
200
100
200
(1, 2, 3, 4, 5, 6, 7, 8, 9)
{'av': 2, 'ab': 4}
```

Lab – 8

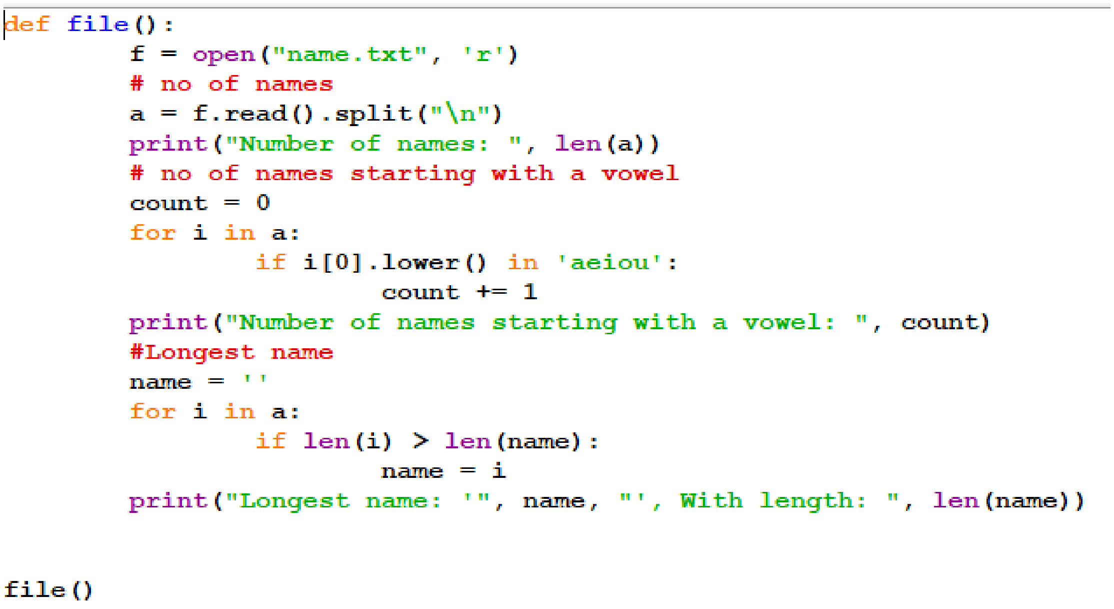
1) Add few names, one name in each row, in “name.txt file”.

a. Count no of names

b. Count all names starting with vowel

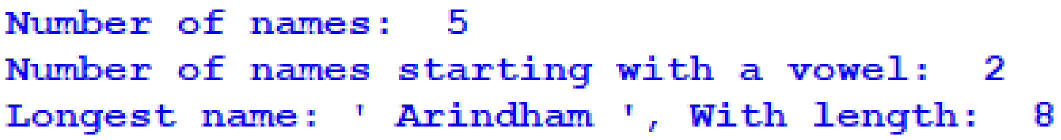
c. Find longest name

Ans:



```
def file():
        f = open("name.txt", 'r')
        # no of names
        a = f.read().split("\n")
        print("Number of names: ", len(a))
        # no of names starting with a vowel
        count = 0
        for i in a:
                if i[0].lower() in 'aeiou':
                        count += 1
        print("Number of names starting with a vowel: ", count)
        #Longest name
        name = ''
        for i in a:
                if len(i) > len(name):
                        name = i
        print("Longest name: '", name, "', With length: ", len(name))


file()
```

Output:



```
Number of names:  5
Number of names starting with a vowel:  2
Longest name: ' Arindham ', With length:  8
```

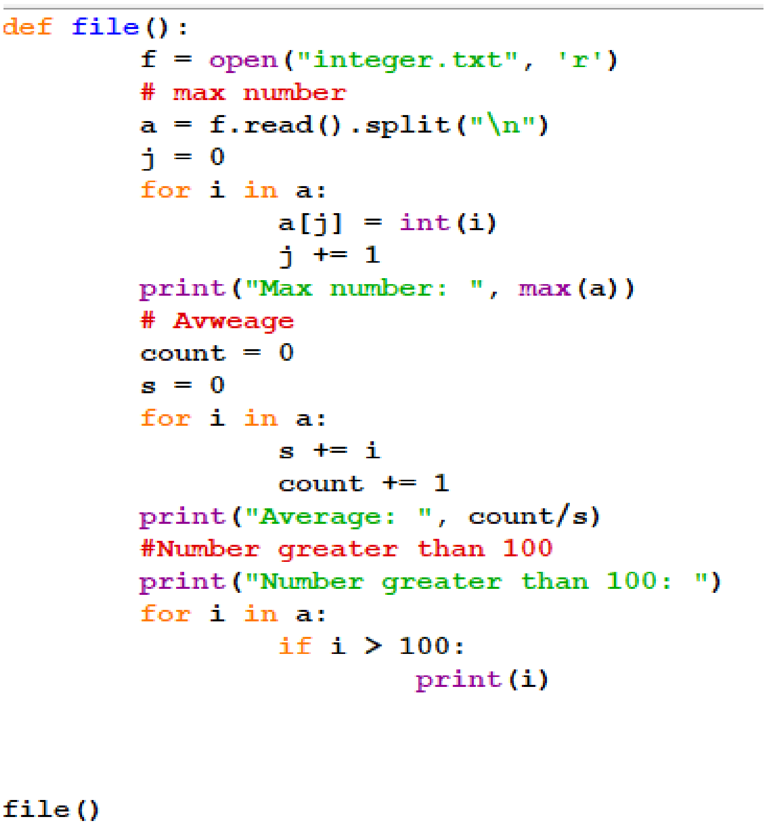
2) Store integers in a file.

a. Find the max number

b. Find average of all numbers

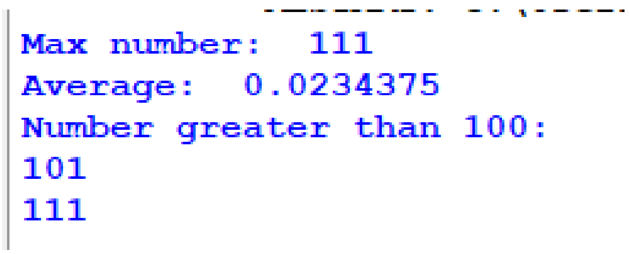
c. Count number of numbers greater than 100

Ans:



```
def file():
        f = open("integer.txt", 'r')
        # max number
        a = f.read().split("\n")
        j = 0
        for i in a:
                a[j] = int(i)
                j += 1
        print("Max number: ", max(a))
        # Avweage
        count = 0
        s = 0
        for i in a:
                s += i
                count += 1
        print("Average: ", count/s)
        #Number greater than 100
        print("Number greater than 100: ")
        for i in a:
                if i > 100:
                        print(i)



file()
```

Output:



```
Max number:  111
Average:  0.0234375
Number greater than 100:
101
111
```

3) Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM)):

Example:

Dehradun 5.78 308.20

Delhi 190 1484

……………

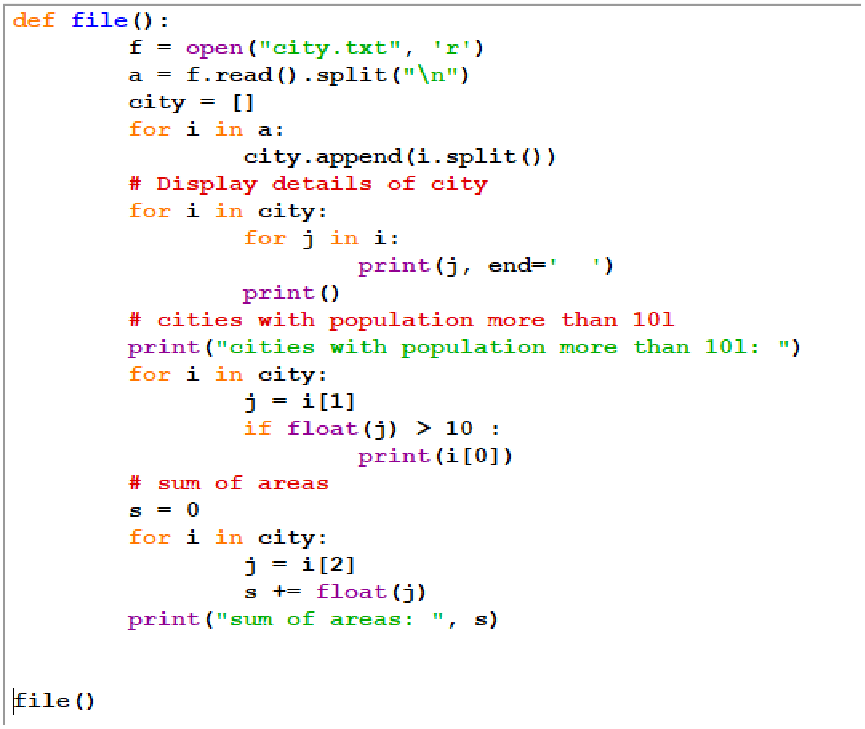
Open file city.txt and read to:

a. Display details of all cities

b. Display city names with population more than 10Lakhs

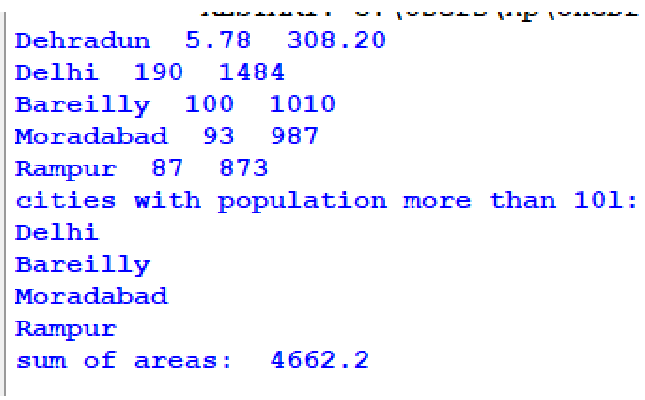
c. Display sum of areas of all cities

Ans:



```
def file():
        f = open("city.txt", 'r')
        a = f.read().split("\n")
        city = []
        for i in a:
                city.append(i.split())
        # Display details of city
        for i in city:
                for j in i:
                        print(j, end='  ')
                print()
        # cities with population more than 10l
        print("cities with population more than 10l: ")
        for i in city:
                j = i[1]
                if float(j) > 10 :
                        print(i[0])
        # sum of areas
        s = 0
        for i in city:
                j = i[2]
                s += float(j)
        print("sum of areas: ", s)


file()
```

Output:



```
Dehradun  5.78  308.20
Delhi  190  1484
Bareilly  100  1010
Moradabad  93  987
Rampur  87  873
cities with population more than 101:
Delhi
Bareilly
Moradabad
Rampur
sum of areas:  4662.2
```

4) Input two values from user where the first line contains N, the number of test cases. The next N lines contain the

space separated values of a and b. Perform integer division and print a/b. Handle exception in case of

ZeroDivisionError or ValueError.

Sample input

1 0

2 $

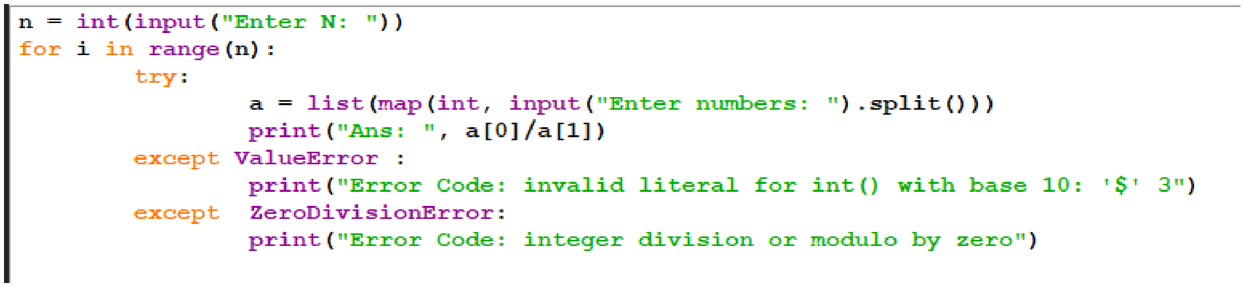
3 1

Sample Output :

Error Code: integer division or modulo by zero

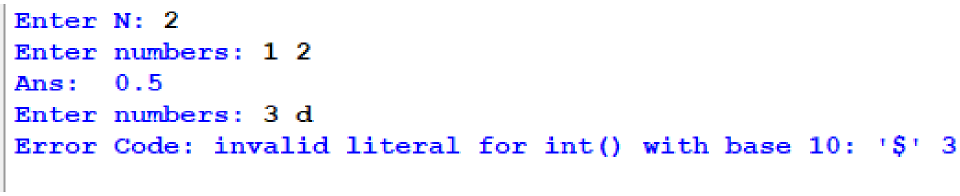
Error Code: invalid literal for int() with base 10: '$' 3

Ans:



```
n = int(input("Enter N: "))
for i in range(n):
        try:
                a = list(map(int, input("Enter numbers: ").split()))
                print("Ans: ", a[0]/a[1])
        except ValueError :
                print("Error Code: invalid literal for int() with base 10: '$' 3")
        except  ZeroDivisionError:
                print("Error Code: integer division or modulo by zero")
```

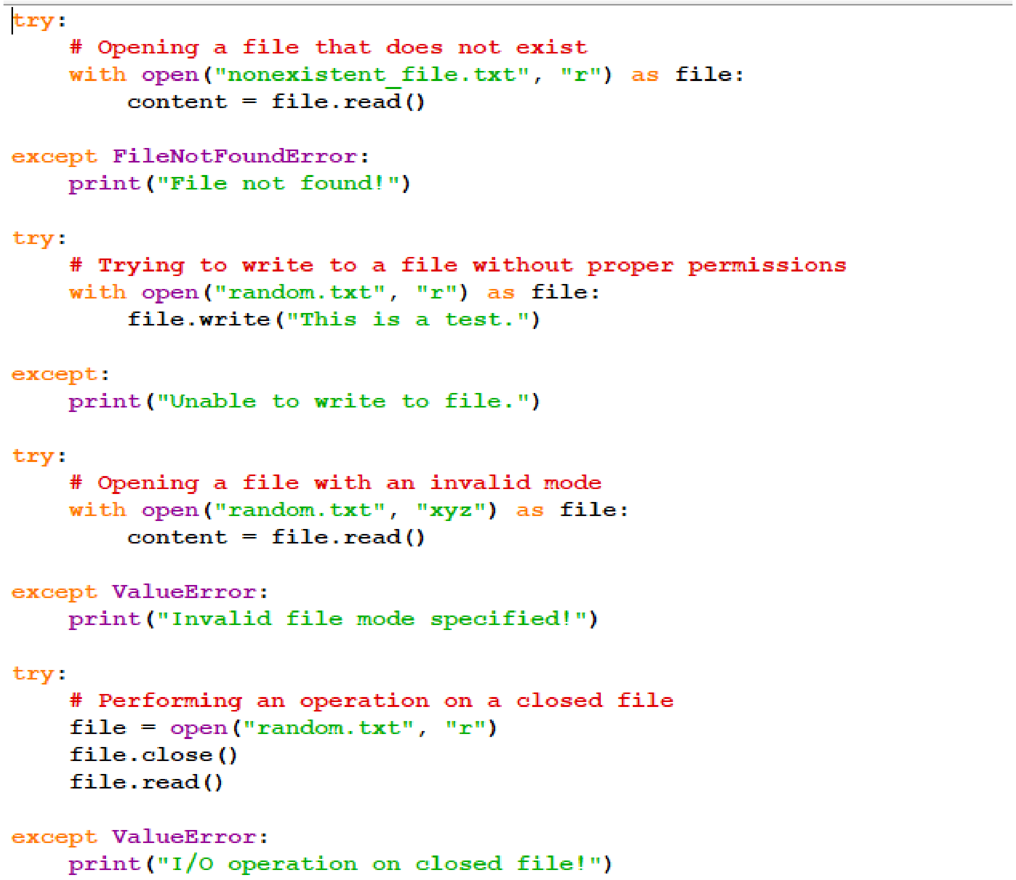
Output:



```
Enter N:  2
Enter numbers:  1 2
Ans:   0.5
Enter numbers:  3 d
Error Code: invalid literal for int() with base 10: '$' 3
```

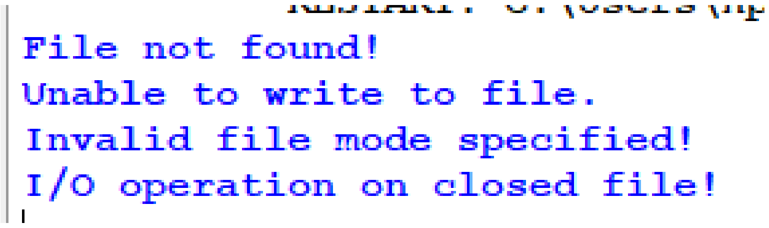
5) Create multiple suitable exceptions for a file handling program.

Ans:



```
try:
    # Opening a file that does not exist
    with open("nonexistent_file.txt", "r") as file:
        content = file.read()

except FileNotFoundError:
    print("File not found!")

try:
    # Trying to write to a file without proper permissions
    with open("random.txt", "r") as file:
        file.write("This is a test.")

except:
    print("Unable to write to file.")

try:
    # Opening a file with an invalid mode
    with open("random.txt", "xyz") as file:
        content = file.read()

except ValueError:
    print("Invalid file mode specified!")

try:
    # Performing an operation on a closed file
    file = open("random.txt", "r")
    file.close()
    file.read()

except ValueError:
    print("I/O operation on closed file!")
```

Output:

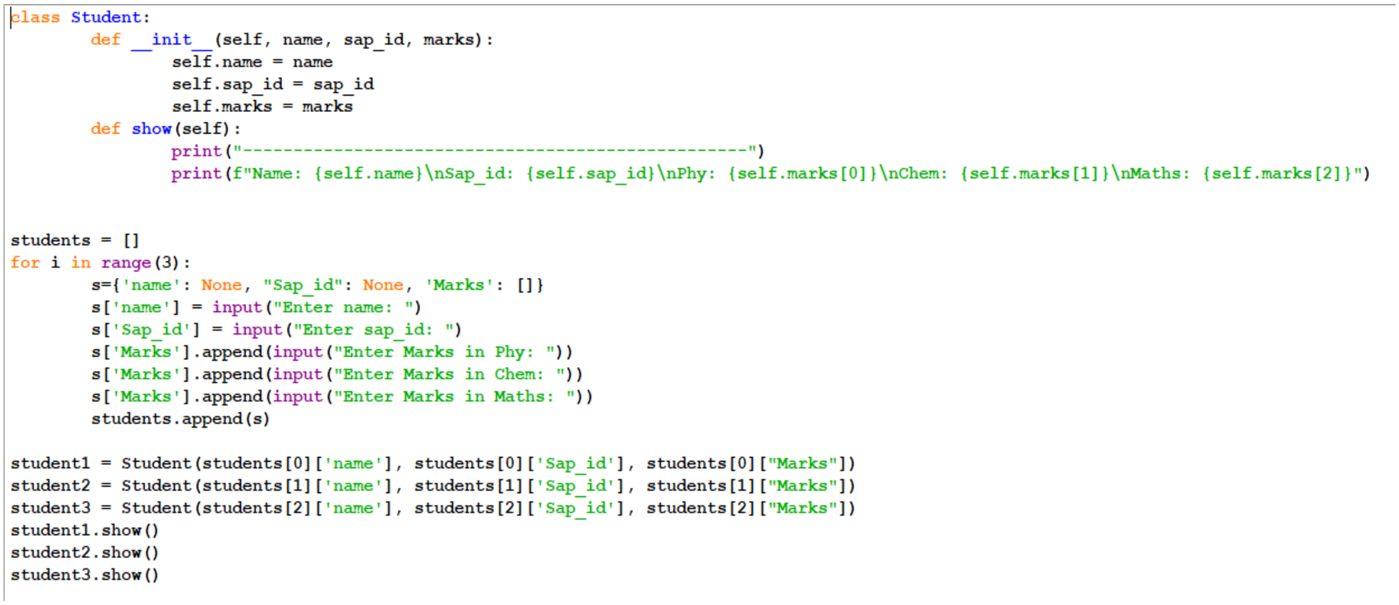


```
File not found!
Unable to write to file.
Invalid file mode specified!
I/O operation on closed file!
```

Lab – 9

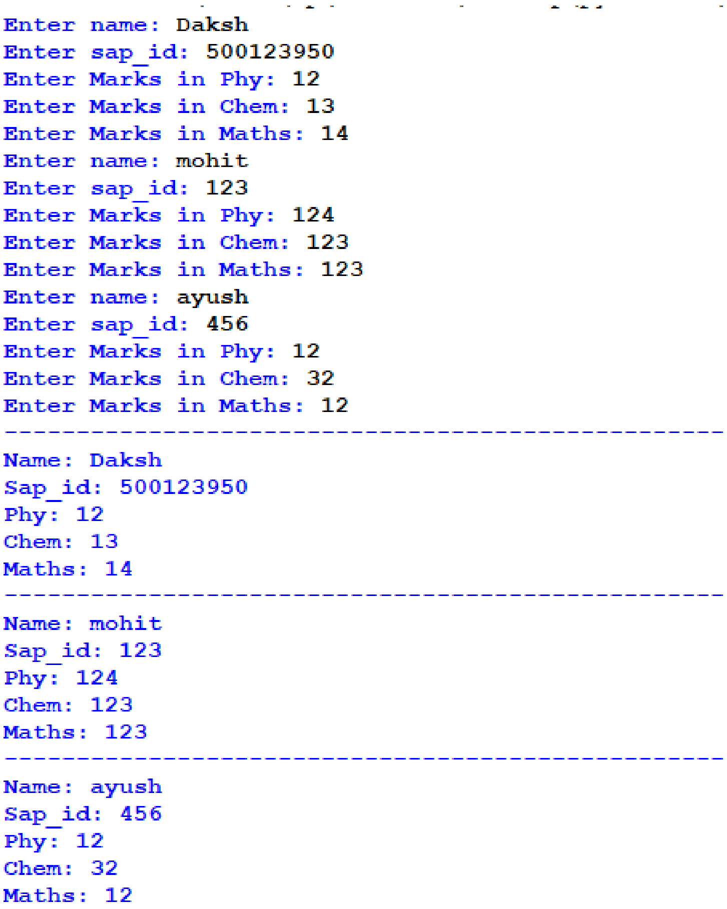
1) Create a class of student (name, sap id, marks[phy,chem,maths]). Create 3 objects by taking inputs from the user and display details of all students.

Ans:



```
class Student:
        def __init__(self, name, sap_id, marks):
                self.name = name
                self.sap_id = sap_id
                self.marks = marks
        def show(self):
                print("--------------------------------------------------")
                print(f"Name: {self.name}\nSap_id: {self.sap_id}\nPhy: {self.marks[0]}\nChem: {self.marks[1]}\nMaths: {self.marks[2]}")


students = []
for i in range(3):
        s={'name': None, "Sap_id": None, 'Marks': []}
        s['name'] = input("Enter name: ")
        s['Sap_id'] = input("Enter sap_id: ")
        s['Marks'].append(input("Enter Marks in Phy: "))
        s['Marks'].append(input("Enter Marks in Chem: "))
        s['Marks'].append(input("Enter Marks in Maths: "))
        students.append(s)

student1 = Student(students[0]['name'], students[0]['Sap_id'], students[0]["Marks"])
student2 = Student(students[1]['name'], students[1]['Sap_id'], students[1]["Marks"])
student3 = Student(students[2]['name'], students[2]['Sap_id'], students[2]["Marks"])
student1.show()
student2.show()
student3.show()
```

Output:



```
Enter name: Daksh
Enter sap_id: 500123950
Enter Marks in Phy: 12
Enter Marks in Chem: 13
Enter Marks in Maths: 14
Enter name: mohit
Enter sap_id: 123
Enter Marks in Phy: 124
Enter Marks in Chem: 123
Enter Marks in Maths: 123
Enter name: ayush
Enter sap_id: 456
Enter Marks in Phy: 12
Enter Marks in Chem: 32
Enter Marks in Maths: 12
--------------------------------------------------
Name: Daksh
Sap_id: 500123950
Phy: 12
Chem: 13
Maths: 14
--------------------------------------------------
Name: mohit
Sap_id: 123
Phy: 124
Chem: 123
Maths: 123
--------------------------------------------------
Name: ayush
Sap_id: 456
Phy: 12
Chem: 32
Maths: 12
```

2) Add constructor in the above class to initialize student details of n students and implement following methods:

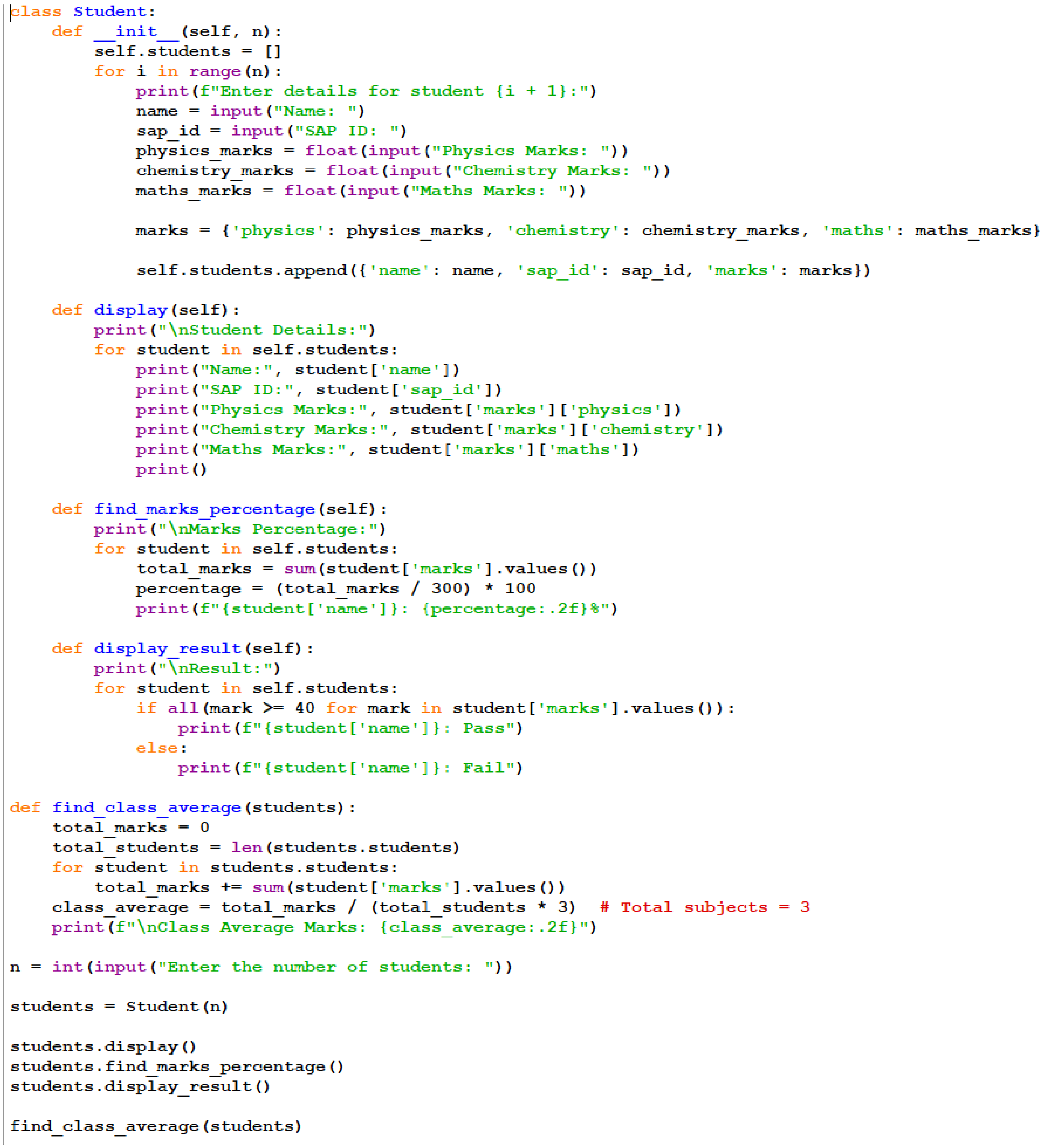
a. Display() student details

b. Find Marks_percentage() of each student

c. Display result() [Note: if marks in each subject >40% than Pass else Fail]

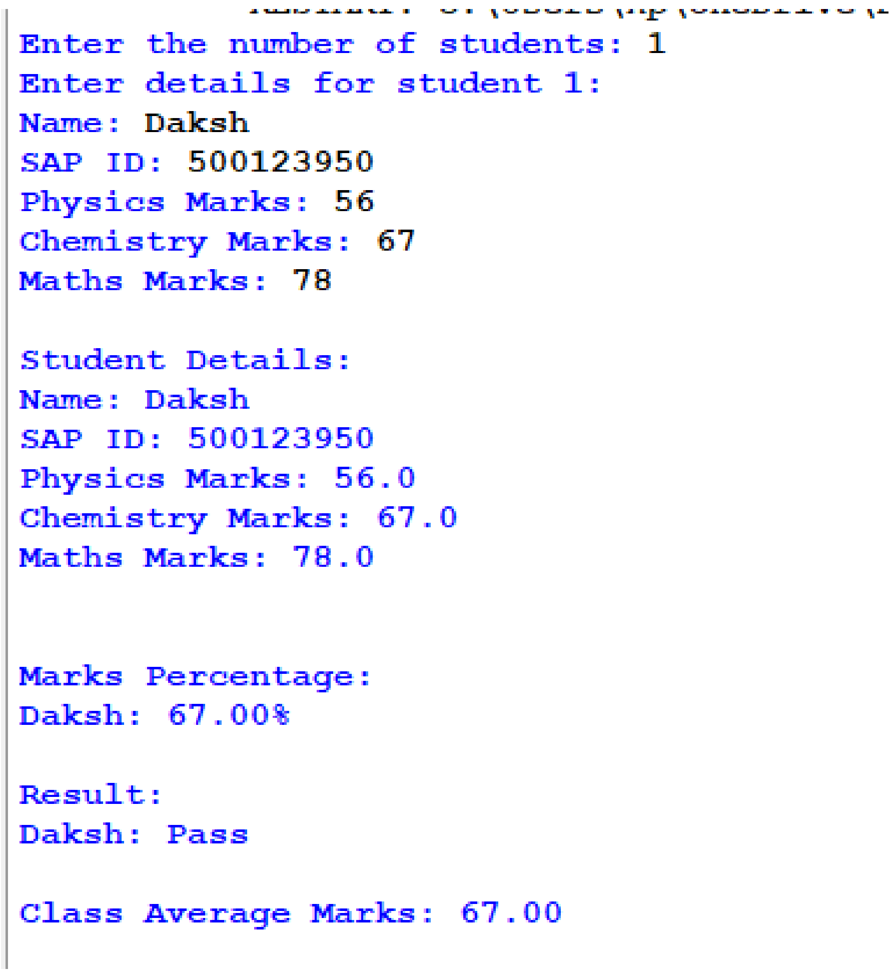
Write a Function to find average of the class.

Ans:



```
class Student:
    def __init__(self, n):
        self.students = []
        for i in range(n):
            print(f"Enter details for student {i + 1}:")
            name = input("Name: ")
            sap_id = input("SAP ID: ")
            physics_marks = float(input("Physics Marks: "))
            chemistry_marks = float(input("Chemistry Marks: "))
            maths_marks = float(input("Maths Marks: "))

            marks = {'physics': physics_marks, 'chemistry': chemistry_marks, 'maths': maths_marks}

            self.students.append({'name': name, 'sap_id': sap_id, 'marks': marks})

    def display(self):
        print("\nStudent Details:")
        for student in self.students:
            print("Name:", student['name'])
            print("SAP ID:", student['sap_id'])
            print("Physics Marks:", student['marks']['physics'])
            print("Chemistry Marks:", student['marks']['chemistry'])
            print("Maths Marks:", student['marks']['maths'])
            print()

    def find_marks_percentage(self):
        print("\nMarks Percentage:")
        for student in self.students:
            total_marks = sum(student['marks'].values())
            percentage = (total_marks / 300) * 100
            print(f"{student['name']}: {percentage:.2f}%")

    def display_result(self):
        print("\nResult:")
        for student in self.students:
            if all(mark >= 40 for mark in student['marks'].values()):
                print(f"{student['name']}: Pass")
            else:
                print(f"{student['name']}: Fail")

def find_class_average(students):
    total_marks = 0
    total_students = len(students.students)
    for student in students.students:
        total_marks += sum(student['marks'].values())
    class_average = total_marks / (total_students * 3)  # Total subjects = 3
    print(f"\nClass Average Marks: {class_average:.2f}")

n = int(input("Enter the number of students: "))

students = Student(n)

students.display()
students.find_marks_percentage()
students.display_result()

find_class_average(students)
```

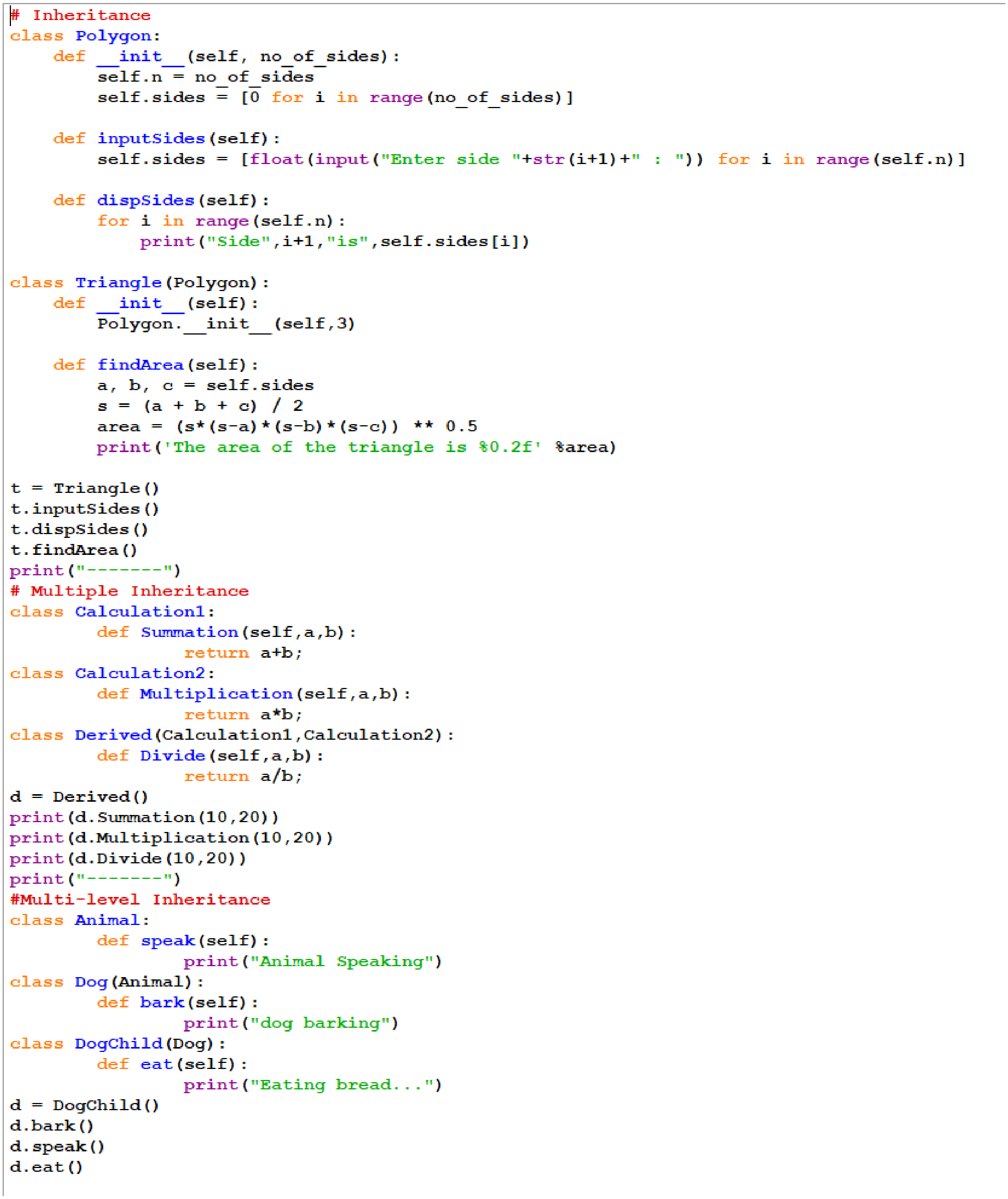
Output:



```
Enter the number of students: 1
Enter details for student 1:
Name: Daksh
SAP ID: 500123950
Physics Marks: 56
Chemistry Marks: 67
Maths Marks: 78

Student Details:
Name: Daksh
SAP ID: 500123950
Physics Marks: 56.0
Chemistry Marks: 67.0
Maths Marks: 78.0


Marks Percentage:
Daksh: 67.00%

Result:
Daksh: Pass

Class Average Marks: 67.00
```

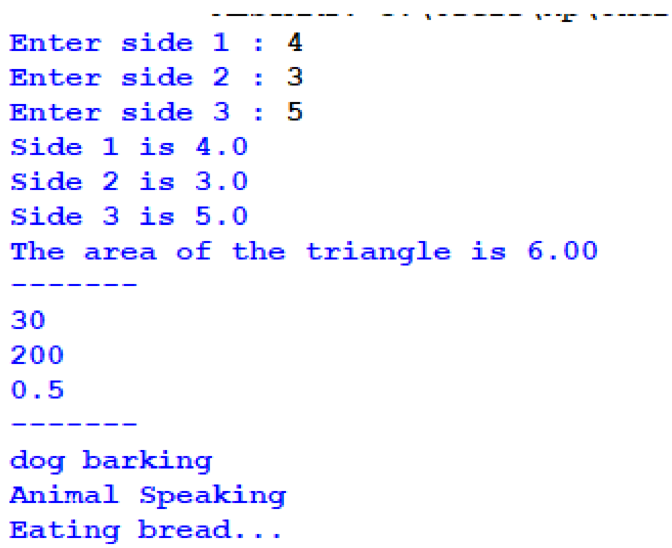
3) Create programs to implement different types of inheritances.

Ans:



```
# Inheritance
class Polygon:
    def __init__(self, no_of_sides):
        self.n = no_of_sides
        self.sides = [0 for i in range(no_of_sides)]

    def inputSides(self):
        self.sides = [float(input("Enter side "+str(i+1)+" : ")) for i in range(self.n)]

    def dispSides(self):
        for i in range(self.n):
            print("Side",i+1,"is",self.sides[i])

class Triangle(Polygon):
    def __init__(self):
        Polygon.__init__(self,3)

    def findArea(self):
        a, b, c = self.sides
        s = (a + b + c) / 2
        area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
        print('The area of the triangle is %0.2f' %area)

t = Triangle()
t.inputSides()
t.dispSides()
t.findArea()
print("-------")
# Multiple Inheritance
class Calculation1:
        def Summation(self,a,b):
                return a+b;
class Calculation2:
        def Multiplication(self,a,b):
                return a*b;
class Derived(Calculation1,Calculation2):
        def Divide(self,a,b):
                return a/b;
d = Derived()
print(d.Summation(10,20))
print(d.Multiplication(10,20))
print(d.Divide(10,20))
print("-------")
#Multi-level Inheritance
class Animal:
        def speak(self):
                print("Animal Speaking")
class Dog(Animal):
        def bark(self):
                print("dog barking")
class DogChild(Dog):
        def eat(self):
                print("Eating bread...")
d = DogChild()
d.bark()
d.speak()
d.eat()
```

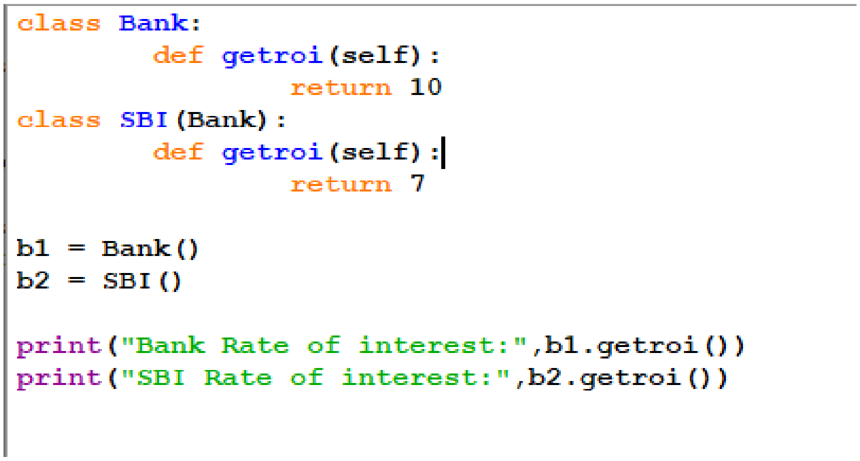
Output:



```
Enter side 1 : 4
Enter side 2 : 3
Enter side 3 : 5
Side 1 is 4.0
Side 2 is 3.0
Side 3 is 5.0
The area of the triangle is 6.00
-------
30
200
0.5
-------
dog barking
Animal Speaking
Eating bread...
```

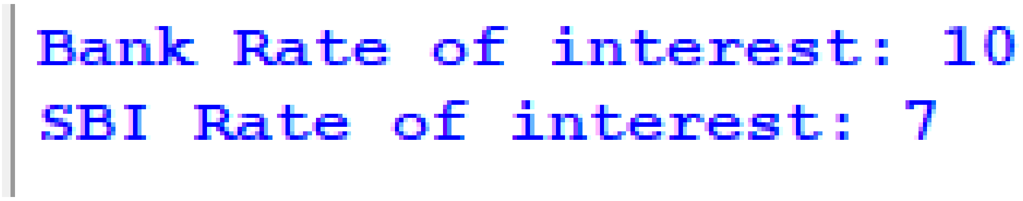
4) Create a class to implement method Overriding.

Ans:



```
class Bank:
        def getroi(self):
                return 10
class SBI(Bank):
        def getroi(self):
                return 7

b1 = Bank()
b2 = SBI()

print("Bank Rate of interest:",b1.getroi())
print("SBI Rate of interest:",b2.getroi())
```

Output:



```
Bank Rate of interest: 10
SBI Rate of interest: 7
```

5) Create a class for operator overloading which adds two Point Objects where Point has x & y values

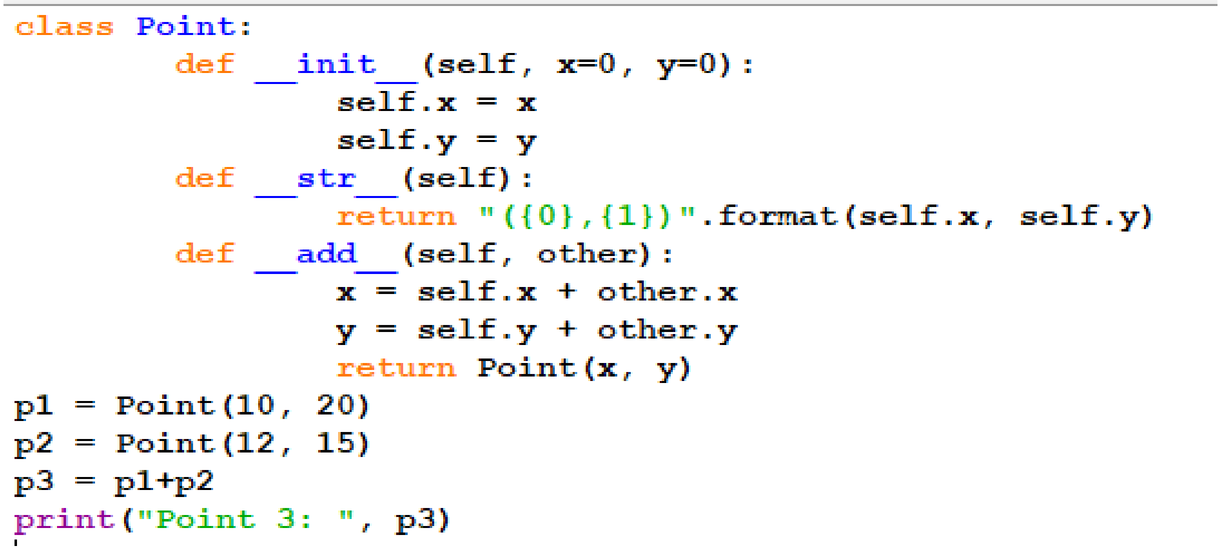
e.g. if

P1(x=10,y=20)

P2(x=12,y=15)

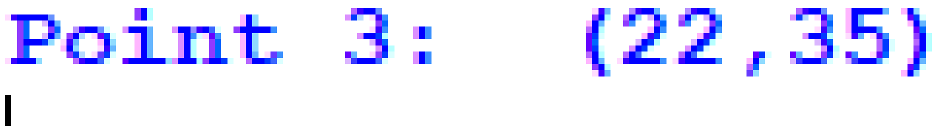
P3=P1+P2 => P3(x=22,y=35)

Ans:



```
class Point:
        def __init__(self, x=0, y=0):
                self.x = x
                self.y = y
        def __str__(self):
                return "({0},{1})".format(self.x, self.y)
        def __add__(self, other):
                x = self.x + other.x
                y = self.y + other.y
                return Point(x, y)
p1 = Point(10, 20)
p2 = Point(12, 15)
p3 = p1+p2
print("Point 3: ", p3)
```

Output:

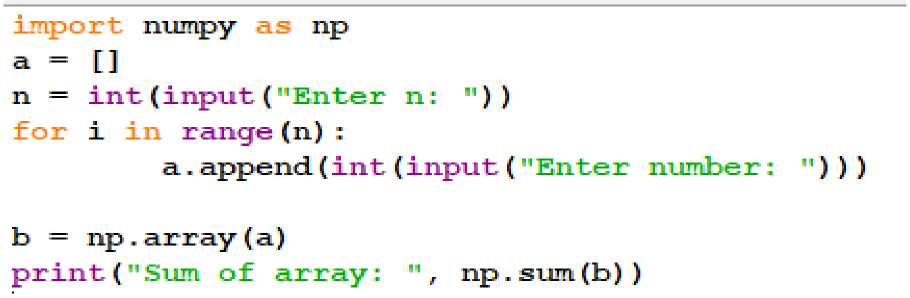


```
Point 3:  (22,35)
```

Lab – 10

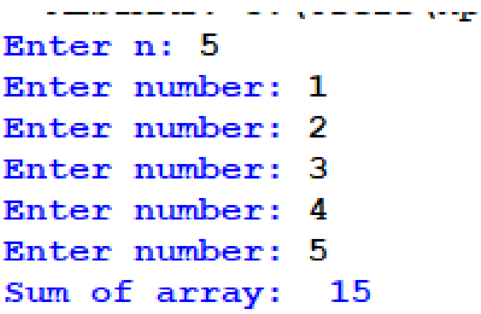
1) Create numpy array to find sum of all elements in an array.

Ans:



```
import numpy as np
a = []
n = int(input("Enter n: "))
for i in range(n):
        a.append(int(input("Enter number: ")))

b = np.array(a)
print("Sum of array: ", np.sum(b))
```

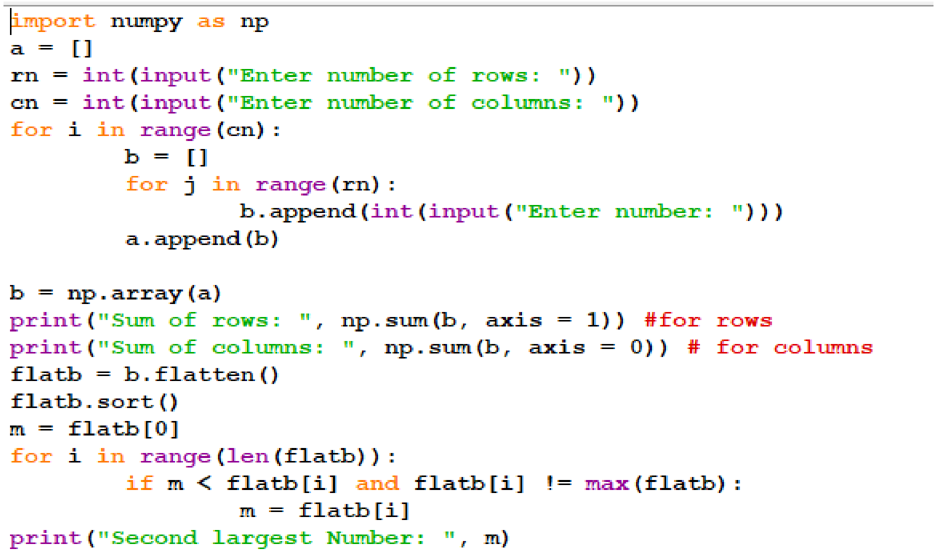
Output:



```
Enter n: 5
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
Enter number: 5
Sum of array:  15
```

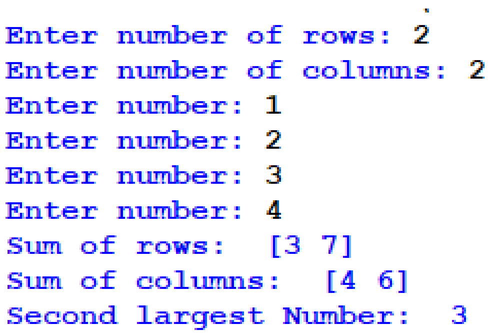
2) Create numpy array of (3,3) dimension. Now find sum of all rows & columns individually. Also find 2nd maximum element in the array.

Ans:



```
import numpy as np
a = []
rn = int(input("Enter number of rows: "))
cn = int(input("Enter number of columns: "))
for i in range(cn):
        b = []
        for j in range(rn):
                b.append(int(input("Enter number: ")))
        a.append(b)

b = np.array(a)
print("Sum of rows: ", np.sum(b, axis = 1)) #for rows
print("Sum of columns: ", np.sum(b, axis = 0)) # for columns
flatb = b.flatten()
flatb.sort()
m = flatb[0]
for i in range(len(flatb)):
        if m < flatb[i] and flatb[i] != max(flatb):
                m = flatb[i]
print("Second largest Number: ", m)
```

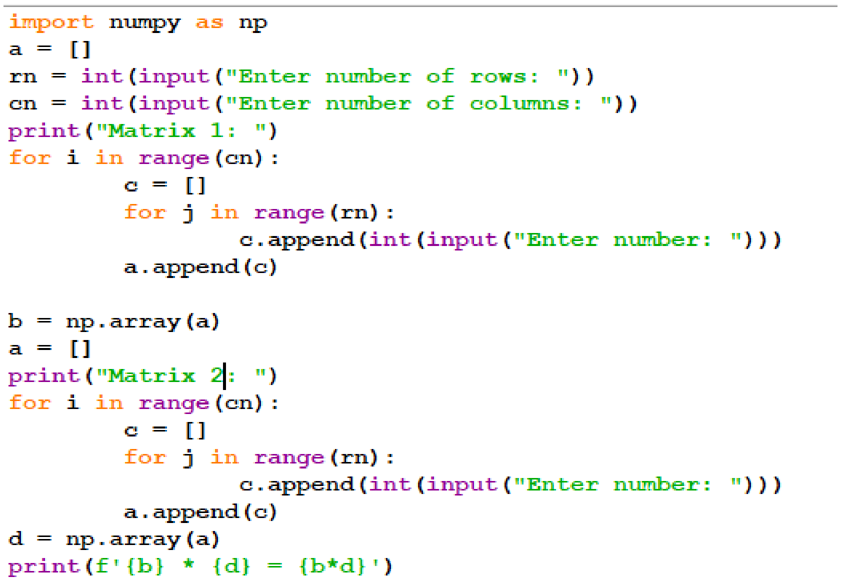
Output:



```
Enter number of rows: 2
Enter number of columns: 2
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
Sum of rows:  [3 7]
Sum of columns:  [4 6]
Second largest Number:  3
```

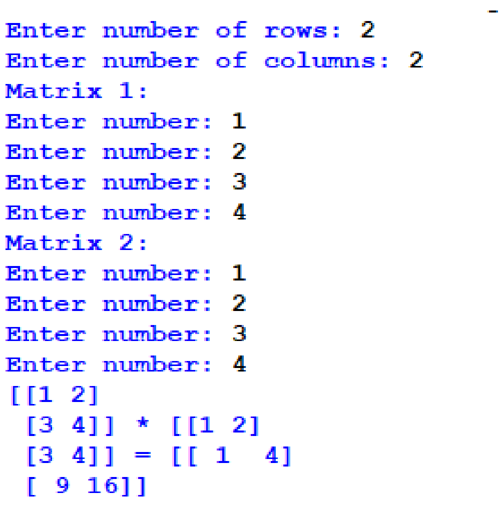
3) Perform Matrix multiplication of any 2 n*n matrices.

Ans:



```
import numpy as np
a = []
rn = int(input("Enter number of rows: "))
cn = int(input("Enter number of columns: "))
print("Matrix 1: ")
for i in range(cn):
        c = []
        for j in range(rn):
                c.append(int(input("Enter number: ")))
        a.append(c)

b = np.array(a)
a = []
print("Matrix 2: ")
for i in range(cn):
        c = []
        for j in range(rn):
                c.append(int(input("Enter number: ")))
        a.append(c)
d = np.array(a)
print(f'{b} * {d} = {b*d}')
```

Output:



```
Enter number of rows: 2
Enter number of columns: 2
Matrix 1:
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
Matrix 2:
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
[[1 2]
 [3 4]] * [[1 2]
 [3 4]] = [[ 1  4]
 [ 9 16]]
```

4) Write a Pandas program to get the powers of an array values element-wise.

Note: First array elements raised to powers from second array

Sample data: {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]}

Expected Output:

X Y Z

0 78 84 86

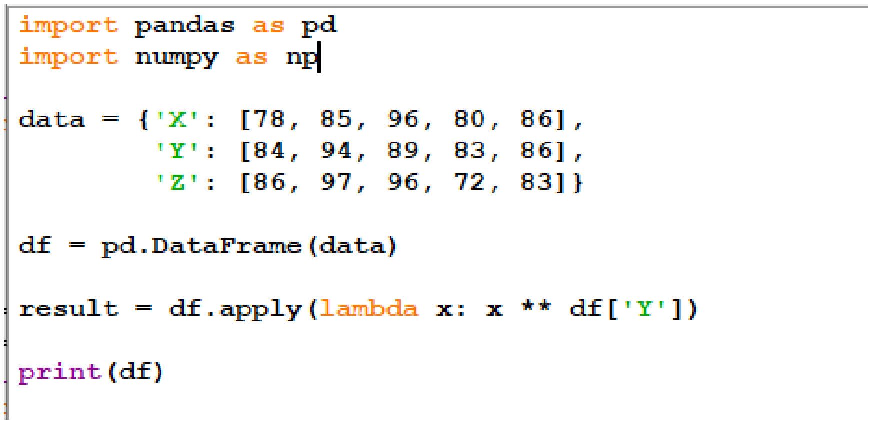
1 85 94 97

2 96 89 96

3 80 83 72

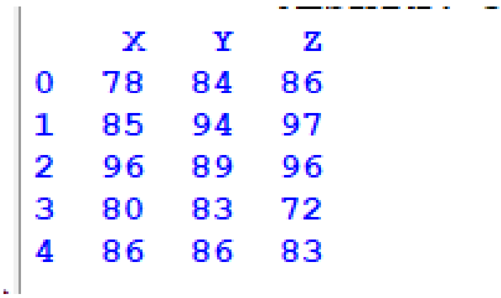
4 86 86 83

Ans:



```
import pandas as pd
import numpy as np

data = {'X': [78, 85, 96, 80, 86],
        'Y': [84, 94, 89, 83, 86],
        'Z': [86, 97, 96, 72, 83]}

df = pd.DataFrame(data)

result = df.apply(lambda x: x ** df['Y'])

print(df)
```

Output:



```
    X   Y   Z
0  78  84  86
1  85  94  97
2  96  89  96
3  80  83  72
4  86  86  83
```

5) Write a Pandas program to get the first 3 rows of a given DataFrame.

Sample Python dictionary data and list labels:

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Expected Output:

First three rows of the data frame:

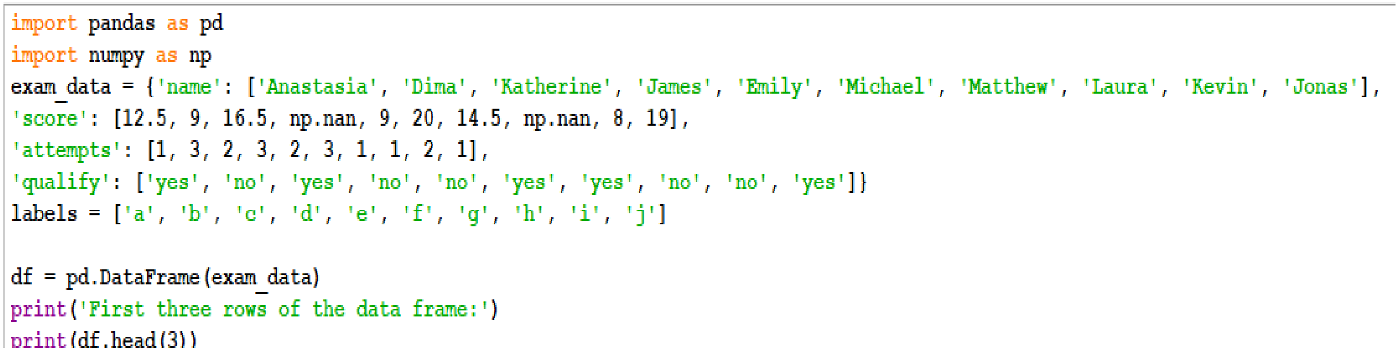
attempts name qualify score

a 1 Anastasia yes 12.5

b 3 Dima no 9.0

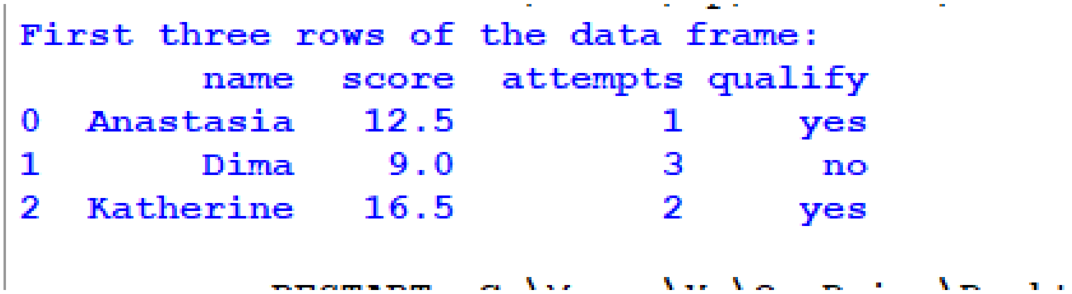
c 2 Katherine yes 16.5

Ans:



```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data)
print('First three rows of the data frame:')
print(df.head(3))
```

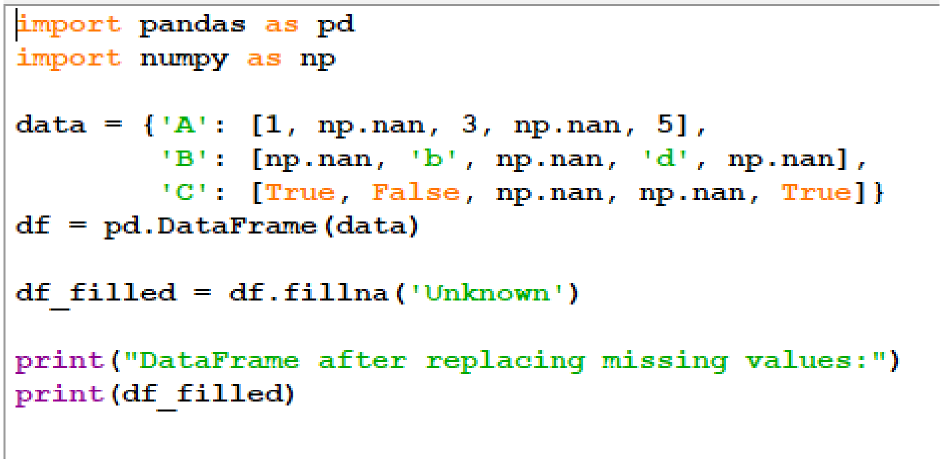
Output:



```
First three rows of the data frame:
        name  score  attempts qualify
0  Anastasia   12.5         1     yes
1       Dima    9.0         3      no
2  Katherine   16.5         2     yes
```

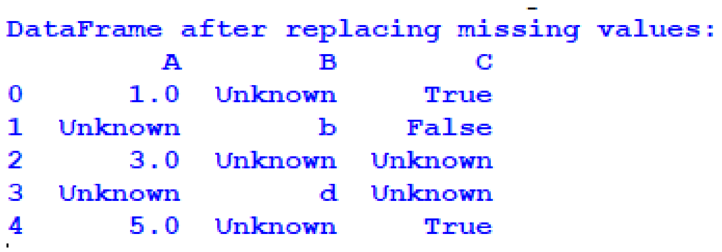
6) Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

Ans:



```
import pandas as pd
import numpy as np

data = {'A': [1, np.nan, 3, np.nan, 5],
        'B': [np.nan, 'b', np.nan, 'd', np.nan],
        'C': [True, False, np.nan, np.nan, True]}
df = pd.DataFrame(data)

df_filled = df.fillna('Unknown')

print("DataFrame after replacing missing values:")
print(df_filled)
```

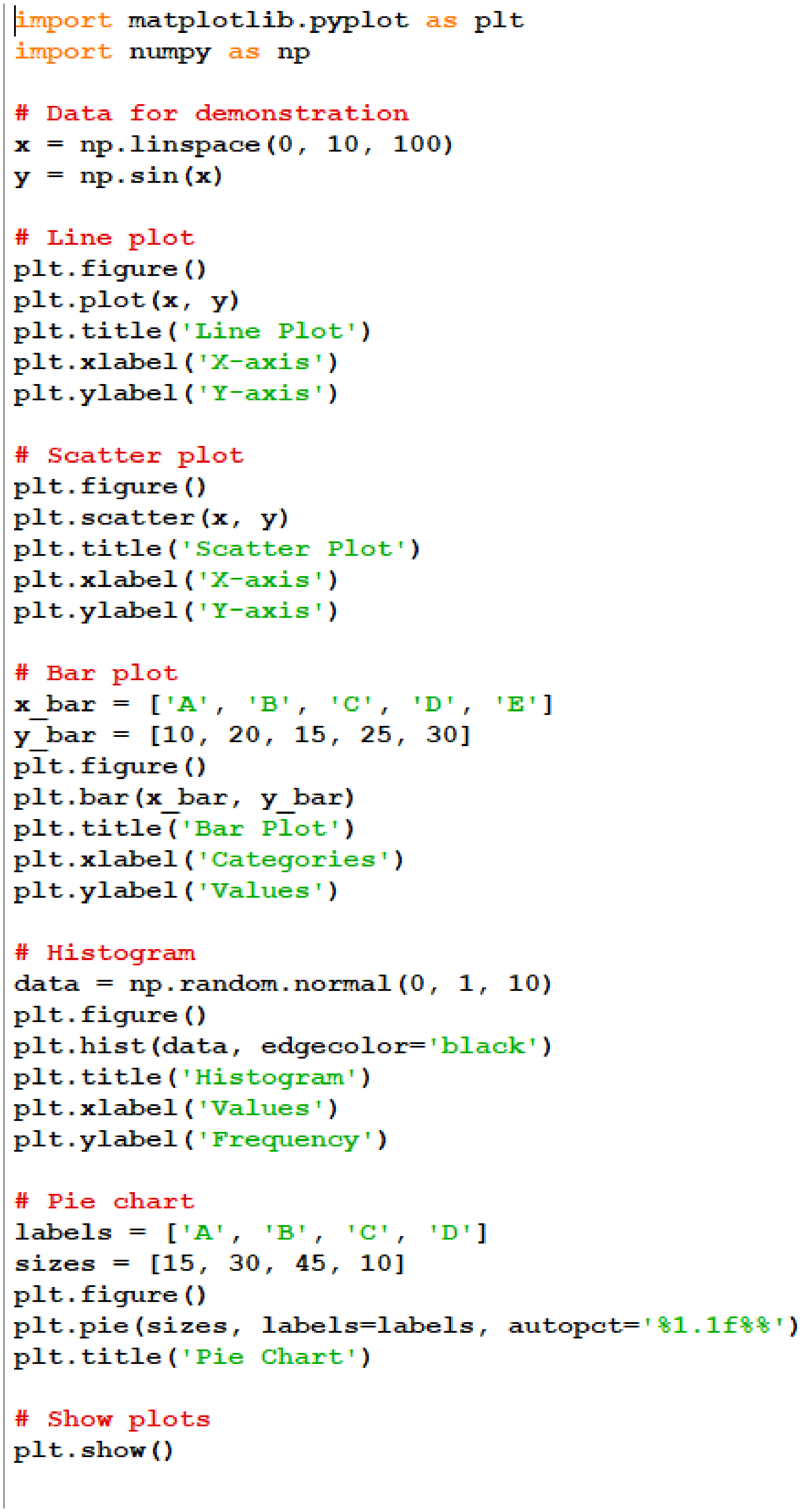
Output:



```
DataFrame after replacing missing values:
         A        B        C
0      1.0  Unknown     True
1  Unknown        b    False
2      3.0  Unknown  Unknown
3  Unknown        d  Unknown
4      5.0  Unknown     True
```

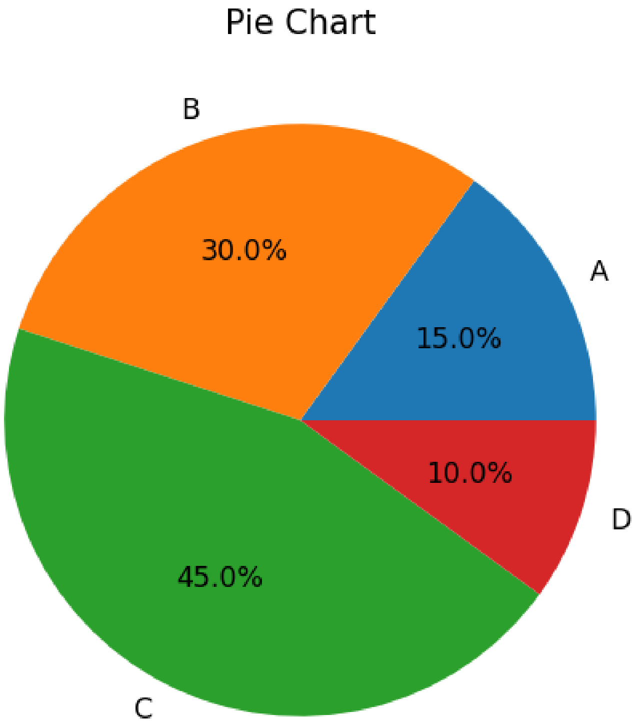
7) Create a program to demonstrate different visual forms using Matplotlib.

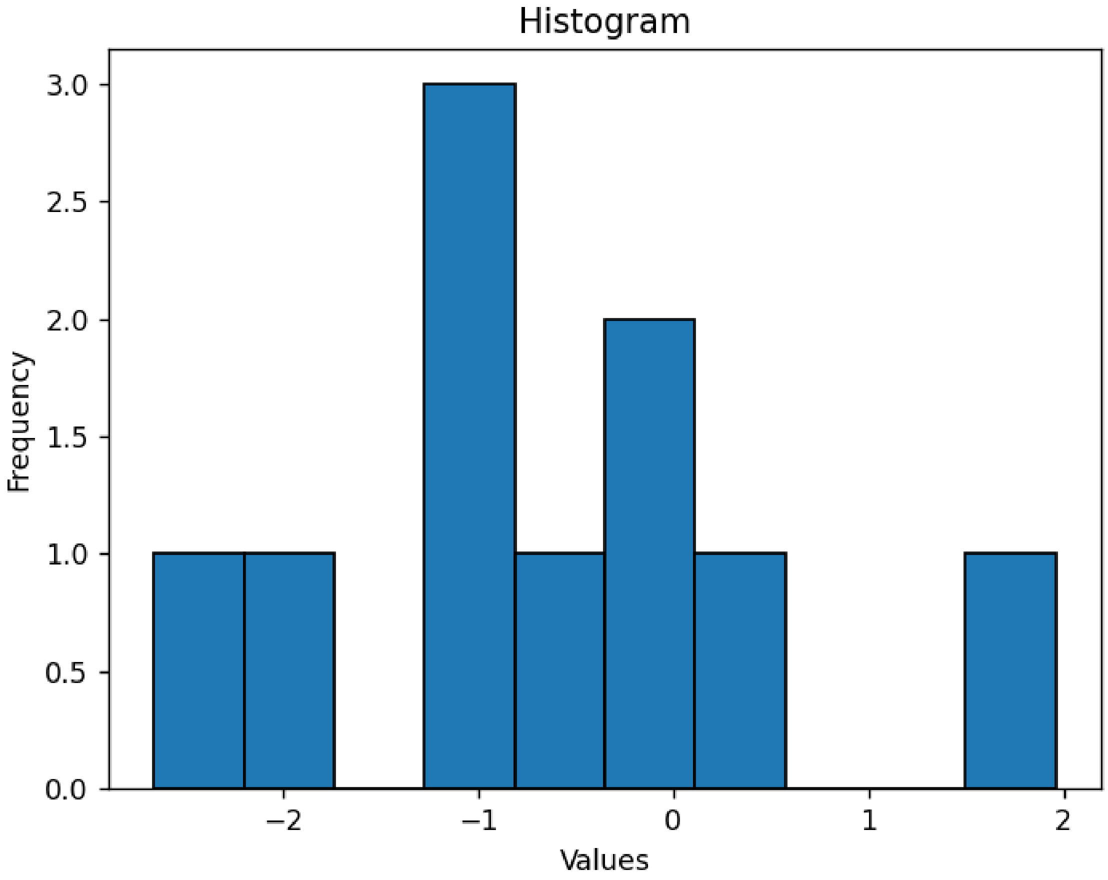
Ans:

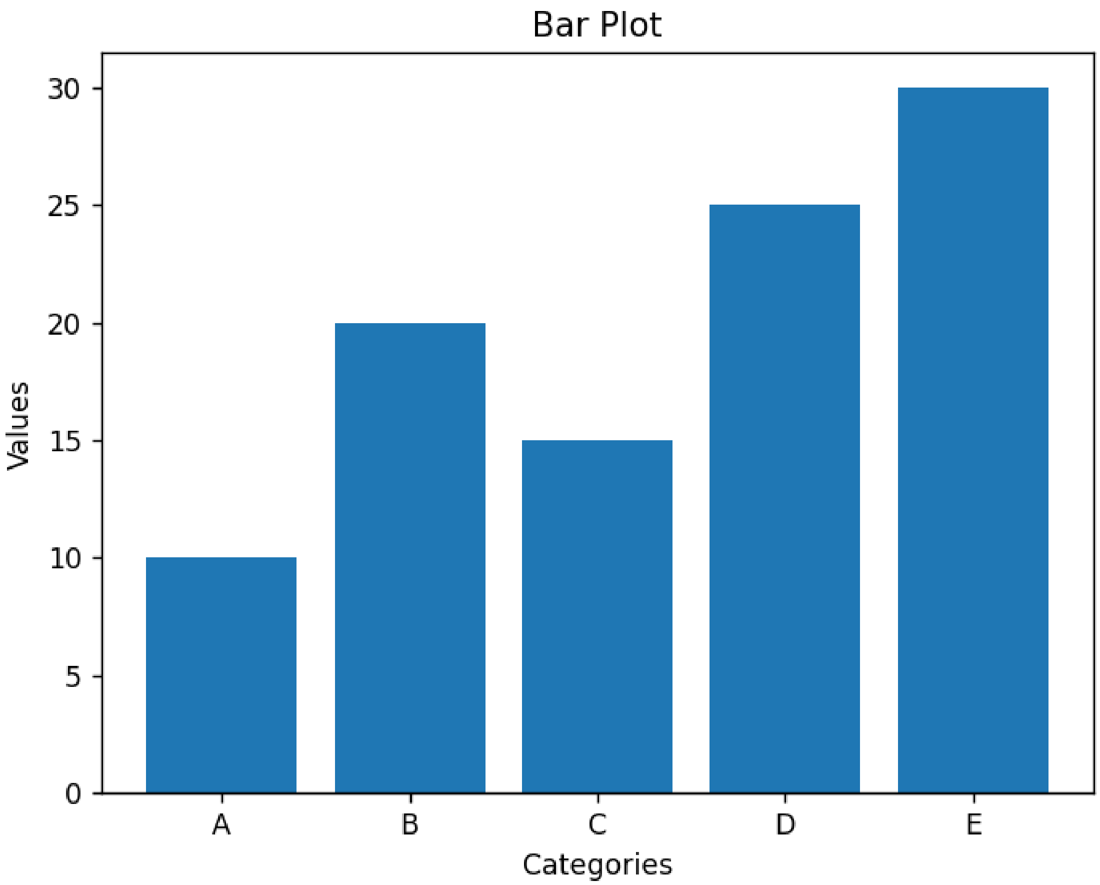


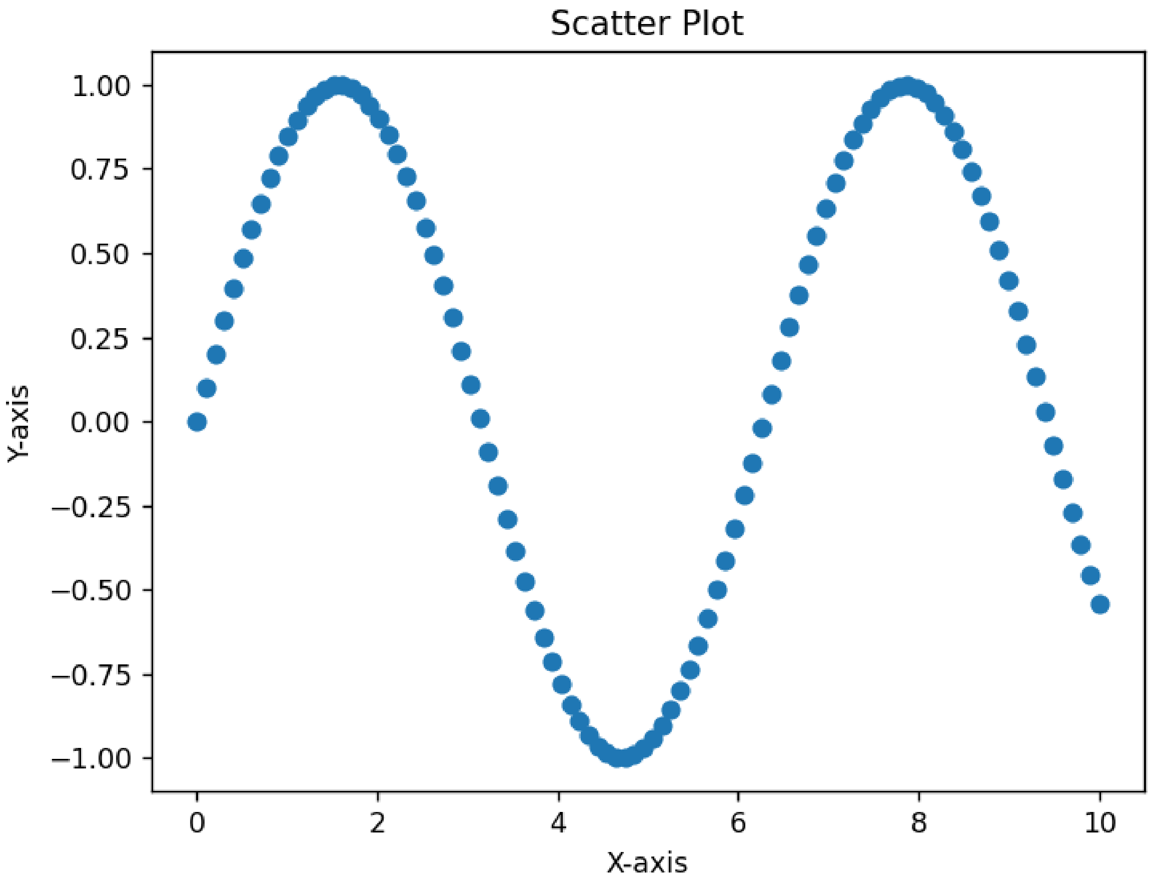
```
import matplotlib.pyplot as plt
import numpy as np

# Data for demonstration
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Line plot
plt.figure()
plt.plot(x, y)
plt.title('Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Scatter plot
plt.figure()
plt.scatter(x, y)
plt.title('Scatter Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Bar plot
x_bar = ['A', 'B', 'C', 'D', 'E']
y_bar = [10, 20, 15, 25, 30]
plt.figure()
plt.bar(x_bar, y_bar)
plt.title('Bar Plot')
plt.xlabel('Categories')
plt.ylabel('Values')

# Histogram
data = np.random.normal(0, 1, 10)
plt.figure()
plt.hist(data, edgecolor='black')
plt.title('Histogram')
plt.xlabel('Values')
plt.ylabel('Frequency')

# Pie chart
labels = ['A', 'B', 'C', 'D']
sizes = [15, 30, 45, 10]
plt.figure()
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title('Pie Chart')

# Show plots
plt.show()
```

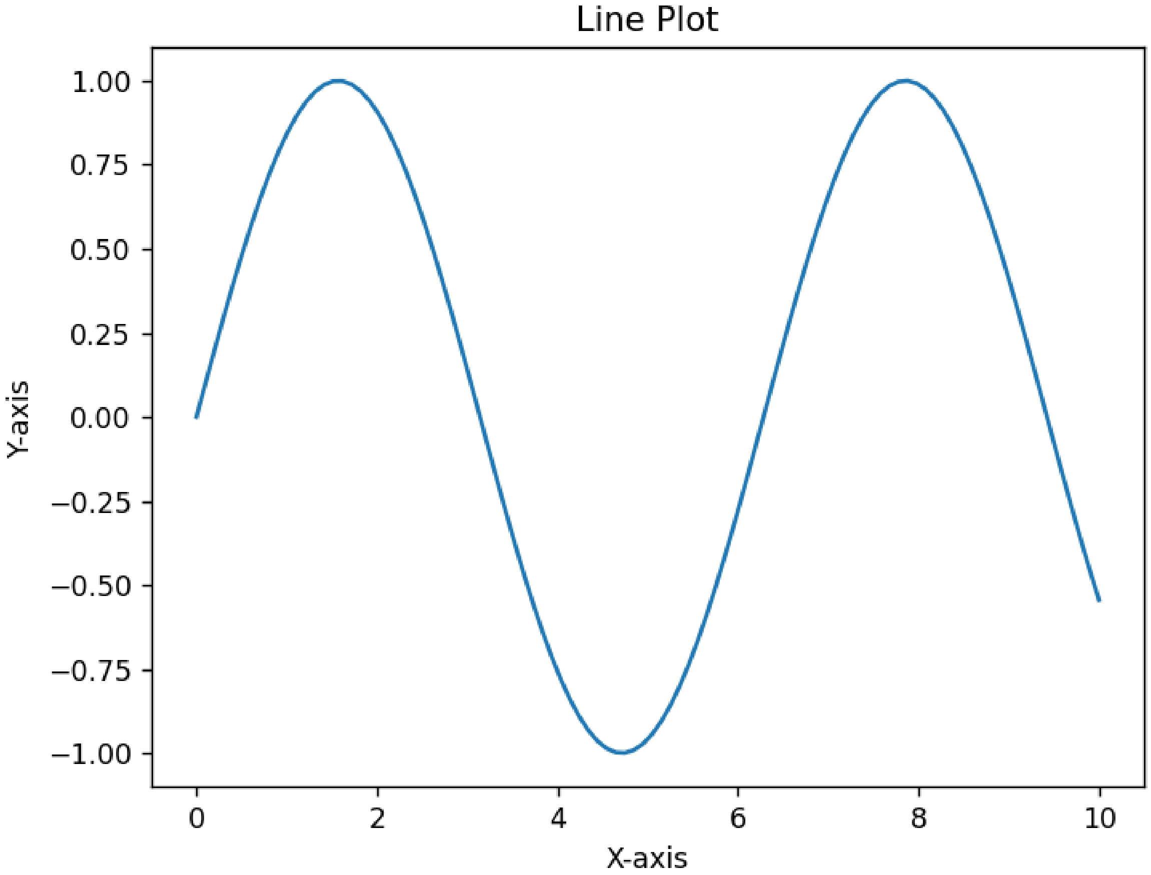
Output:


Pie Chart
B
A
30.0%
15.0%
10.0%
D
45.0%
C


Histogram
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency
−2
−1
0
1
2
Values


Bar Plot
Values
30
25
20
15
10
5
0
A
B
C
D
E
Categories


Scatter Plot
Y-axis
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10
X-axis


Line Plot
Y-axis
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10
X-axis