

S.No: 1	Exp. Name: <i>Write a Program to compute distance between two points taking input from the user (Pythagorean Theorem)</i>	Date: 2023-02-01
---------	--	------------------

Aim:

Write a program to compute distance between two points taking input from the user (Pythagorean Theorem).

Formula for **Pythagorean** theorem for compute distance between two points is:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

At the time of execution, the program should print the following messages one by one on the console as:

Enter the x coordinate of first point:
 Enter the y coordinate of first point:
 Enter the x coordinate of second point:
 Enter the y coordinate of second point:

For example, if the user gives the **input** as:

Enter the x coordinate of first point: 20
 Enter the y coordinate of first point: 22
 Enter the x coordinate of second point: 31
 Enter the y coordinate of second point: 29

then the program should **print** the result as:

Distance is: 13.038404810405298

Source Code:

Lab2a.py

```
x1=int(input("Enter the x coordinate of first point: "))
y1=int(input("Enter the y coordinate of first point: "))
x2=int(input("Enter the x coordinate of second point: "))
y2=int(input("Enter the y coordinate of second point: "))
x=x2-x1
y=y2-y1
x=x**2
y=y**2
d=x+y
d=d**(1/2)
print("Distance is:",d)
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter the x coordinate of first point:

23

Enter the y coordinate of first point:

10

Enter the x coordinate of second point:

25

Enter the y coordinate of second point:

16

Distance is: 6.324555320336759

Test Case - 2

User Output

Enter the x coordinate of first point:

20

Enter the y coordinate of first point:

14

Enter the x coordinate of second point:

13

Enter the y coordinate of second point:

21

Distance is: 9.899494936611665

S.No: 2

Exp. Name: **Write a Program to check whether the given number is an Even number or Odd**

Date: 2023-02-01

Aim:

Write a program to check whether the given number is an even number or odd.

The data type of the return value of input statement is string.

So we need to typecast it to integer after getting the input.

At the time of execution, the program should print the message on the console as:

Enter an integer:

For example, if the user gives the input as:

Enter an integer: 14

then the program should print the result as:

Given number 14 is even

For example, if the user given the input as 141, then the program should print the result as "Given number 141 is odd".

Source Code:

Lab3a.py

```
n=int(input("Enter an integer: "))
if n%2==0:
    print("Given number %d is even"%n)
else:
    print("Given number %d is odd"%n)
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter an integer:

156

Given number 156 is even

Test Case - 2

User Output

Enter an integer:

159

Given number 159 is odd

Test Case - 3

User Output

Enter an integer:

115692

Given number 115692 is even

S.No: 3

Exp. Name: **Write a Program that print out the decimal equivalents of n/1, n/2, n/3, n/4, . . . , n/10 using for loop**

Date: 2023-02-01

Aim:

Write a program to read **n** value, and prints out the decimal equivalents of **n/1, n/2, n/3, n/4, . . . , n/10** using **for loop**.

At the time of execution, the program should print the message on the console as:

Enter n value:

For example, if the user gives the input as:

Enter n value: 1

then the program should print the result as:

```
1/1 = 1.0  
1/2 = 0.5  
1/3 = 0.3333333333333333  
1/4 = 0.25  
1/5 = 0.2  
1/6 = 0.1666666666666666  
1/7 = 0.14285714285714285  
1/8 = 0.125  
1/9 = 0.1111111111111111  
1/10 = 0.1
```

Source Code:

Lab3b.py

```
n=int(input("Enter n value: "))  
for i in range(1,11,1):  
    print("{} / {} = {}".format(n,i,n/i))
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter n value:

4

4/1 = 4.0

4/2 = 2.0

4/3 = 1.3333333333333333

4/4 = 1.0

4/5 = 0.8

4/6 = 0.6666666666666666

4/7 = 0.5714285714285714

4/8 = 0.5

4/9 = 0.4444444444444444

4/10 = 0.4

Test Case - 2

User Output

Enter n value:

6

6/1 = 6.0

6/2 = 3.0

6/3 = 2.0

6/4 = 1.5

6/5 = 1.2

6/6 = 1.0

6/7 = 0.8571428571428571

6/8 = 0.75

6/9 = 0.6666666666666666

6/10 = 0.6

Test Case - 3

User Output

Enter n value:

1

1/1 = 1.0

1/2 = 0.5

1/3 = 0.3333333333333333

1/4 = 0.25

1/5 = 0.2

1/6 = 0.1666666666666666

1/7 = 0.14285714285714285

1/8 = 0.125

1/9 = 0.1111111111111111

1/10 = 0.1

S.No: 4

Exp. Name: **Write a Program to find the Sum of all numbers in List using for loop**

Date: 2023-02-01

Aim:

Write a program to find the **sum** of all numbers in **list** using for loop that loops over a sequence and print **good bye..!** after completion of iteration. Also explain What is sequence?

At the time of execution, the program should print the message on the console as:

Enter size of list:

For example, if the user gives the input as:

Enter size of list: 3

Now, the program should print the following messages one by one on the console as:

Enter value:
Enter value:
Enter value:

For example, if the user gives the input as:

Enter value: 6
Enter value: 3
Enter value: 7

then the program should print the result as:

The sum of the given sequence [6, 3, 7] is 16
good bye..!

Source Code:

Lab3c.py

```
'''k = []      #List declaration
n = int(input("Enter size of list: "))

#Write your code here.....


print("The sum of the given sequence", , "is", )
print("good bye..!")'''

n=int(input("Enter size of list: "))
l=[]
s=0
for i in range(n):
    e=int(input("Enter value: "))
    l.append(e)
    s=s+e
print("The sum of the given sequence {} is {}".format(l,s))
print("good bye..!")
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter size of list:
3
Enter value:
6
Enter value:
3
Enter value:
7
The sum of the given sequence [6, 3, 7] is 16
good bye...!

Test Case - 2
User Output
Enter size of list:
5
Enter value:
22
Enter value:
44
Enter value:
11
Enter value:
55
Enter value:
66
The sum of the given sequence [22, 44, 11, 55, 66] is 198
good bye...!

S.No: 5

Exp. Name: **Write a Program to Print Sequence of numbers of the given number in Decreasing order using While loop**

Date: 2023-08-30

Aim:

Write a program using **while** loop that asks the user for a number and prints a countdown from that number to **zero** (Excluded) and print **good bye..!** after completion of iteration.

Source Code:

Lab3d.py

```
n=int(input("Enter an integer: "))
while(n>0):
    print(n)
    n-=1
print("good bye..!")
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter an integer:

4

4

3

2

1

good bye..!

Test Case - 2

User Output

Enter an integer:

8

8

7

6

5

4

3

2

1

good bye..!

S.No: 6

Exp. Name: **Write a Program to find the Sum of all the Prime numbers of Fibonacci series of n numbers**

Date: 2023-02-01

Aim:

Write a program to find the **sum** of all the **prime** numbers of **fibonacci** series of **n** numbers, where **n** is taken from the input.

Each new term in the fibonacci sequence is generated by adding the previous two terms. By starting with **1** and **2**, the first **10** terms will be:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

At the time of execution, the program should print the message on the console as:

Enter an integer value:

For example, if the user gives the input as:

Enter an integer value: 10

then the program should print the result as:

Fibonacci numbers: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
The prime numbers of fibonacci series: [2, 3, 5, 13]
The sum of the prime numbers: 23

Source Code:

Lab4a.py

```

n=int(input("Enter an integer value: "))
fn=[]
a=-1
b=1
e=1
while(e<=n):
    c=a+b
    fn.append(c)
    a=b
    b=c
    e=e+1
print("Fibonacci numbers:",fn)
flag=0
pn=[]
s=0
for i in fn:
    if(i!=0 and i!=1):
        for j in range(2,i):
            if i%j==0:
                flag=1
        if flag==0:
            pn.append(i)
            s=s+i
    flag=0
print("The prime numbers of fibonacci series:",pn)
print("The sum of the prime numbers:",s)

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter an integer value:

7

Fibonacci numbers: [0, 1, 1, 2, 3, 5, 8]

The prime numbers of fibonacci series: [2, 3, 5]

The sum of the prime numbers: 10

Test Case - 2

User Output

Enter an integer value:

10

Fibonacci numbers: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

The prime numbers of fibonacci series: [2, 3, 5, 13]

The sum of the prime numbers: 23

Test Case - 3

User Output

Enter an integer value:

14

Fibonacci numbers: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233]

The prime numbers of fibonacci series: [2, 3, 5, 13, 89, 233]

The sum of the prime numbers: 345

S.No: 7

Exp. Name: **Write a Program to find the Sum of the Even-valued terms of the Fibonacci sequence**

Date: 2023-02-01

Aim:

By considering the terms in the **Fibonacci sequence** whose values do not exceed a given number, write a program to find the **sum** of the **even-valued** terms.

Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.....

At the time of execution, the program should print the message on the console as:

Enter an integer:

For example, if the user gives the input as:

Enter an integer: 20

then the program should print the result as:

The sum of even numbers of fibonacci sequence [0, 1, 1, 2, 3, 5, 8, 13] is: 10

Source Code:

Lab4b.py

```
n=int(input("Enter an integer: "))
c=s=0
f=[]
a=-1
b=1
while(c<=n):
    c=a+b
    if c<=n:
        f.append(c)
        if c%2==0:
            s=s+c
        a=b
        b=c
print("The sum of even numbers of fibonacci sequence {} is: {}".format(f,s))
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter an integer:

6

The sum of even numbers of fibonacci sequence [0, 1, 1, 2, 3, 5] is: 2

Test Case - 2

User Output

Enter an integer:

20

The sum of even numbers of fibonacci sequence [0, 1, 1, 2, 3, 5, 8, 13] is: 10

S.No: 8

Exp. Name: **Write a Program to Count the number of characters in the String and store them in a Dictionary data structure**

Date: 2023-02-01

Aim:

Write a program to **count** the number of characters in the **string** and store them in a **dictionary** data structure.

At the time of execution, the program should print the message on the console as:

Enter the string:

For example, if the user gives the input as:

Enter the string: Python World

then the program should print the result as:

{'Python World': 12}

Source Code:

Lab5a.py

```
s=input("Enter the string: ")
print("{'%s': %d}%(s,len(s))")
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter the string:

Python

{'Python': 6}

Test Case - 2

User Output

Enter the string:

Hello

{'Hello': 5}

S.No: 9

Exp. Name: **Write a Program to use split() and join() methods to trace Birthday with a Dictionary data structure.**

Date: 2023-08-30

Aim:

Write a program to use split() and join() methods in the **string** and trace the birthday with dictionary data structure. Print the **dictionary** result in the sorted order by using sorted().

Note: Use **\n** at the end of input statement.

Source Code:

Lab5b.py

```
a=input("Enter strings contained name and dob with : seperated\n").split()
dic=dict()
for i in a:
    val=i.split(':')
    dic[val[0]]=val[1]
print("The list:",a)
fin="---".join(a)
print('The list with join:',fin)
print('The sorted dictionary:', sorted(dic.items()))
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter strings contained name and dob with : seperated

Sachin:24/04/1975 Kohli:10/08/1990

Dhoni:18/09/1980

The list: ['Sachin:24/04/1975', 'Kohli:10/08/1990', 'Dhoni:18/09/1980']

The list with join: Sachin:24/04/1975---Kohli:10/08/1990---Dhoni:18/09/1980

The sorted dictionary: [('Dhoni', '18/09/1980'), ('Kohli', '10/08/1990'), ('Sachin', '24/04/1975')]

Test Case - 2

User Output

Enter strings contained name and dob with : seperated

Dhoni:18/09/1980 Sachin:24/04/1975

Kohli:10/08/1990

The list: ['Dhoni:18/09/1980', 'Sachin:24/04/1975', 'Kohli:10/08/1990']

The list with join: Dhoni:18/09/1980---Sachin:24/04/1975---Kohli:10/08/1990

The sorted dictionary: [('Dhoni', '18/09/1980'), ('Kohli', '10/08/1990'), ('Sachin', '24/04/1975')]

S.No: 10

Exp. Name: **Write a Program that combines two Lists into a Dictionary**

Date: 2023-02-01

Aim:

Write a program that **combines** two **lists** into a **dictionary**.

At the time of execution, the program should print the following messages one by one on the console as:

Enter size of int-list:
Enter size of string-list:

For example, if the user gives the **input** as:

Enter size of int-list: 4
Enter size of string-list: 3

Now, the program should print the following messages one by one on the console as:

Enter int for int-list:
Enter int for int-list:
Enter int for int-list:
Enter int for int-list:
Enter string for string-list:
Enter string for string-list:
Enter string for string-list:

For example, if the user gives the **input** as:

Enter int for int-list: 1
Enter int for int-list: 2
Enter int for int-list: 3
Enter int for int-list: 4
Enter string for string-list: Saraswathi
Enter string for string-list: Ganga
Enter string for string-list: Yamuna

then the program should **print** the result as:

{1: 'Saraswathi', 2: 'Ganga', 3: 'Yamuna'}

Source Code:

Lab6a.py

```
ni=int(input("Enter size of int-list: "))  
ns=int(input("Enter size of string-list: "))  
li=[]  
ls=[]  
for i in range(ni):  
    n=int(input("Enter int for int-list: "))  
    li.append(n)  
for i in range(ns):  
    n=input("Enter string for string-list: ")  
    ls.append(n)  
d=dict(zip(li,ls))  
print(d)
```

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	
Enter size of int-list:	
3	
Enter size of string-list:	
3	
Enter int for int-list:	
10	
Enter int for int-list:	
11	
Enter int for int-list:	
12	
Enter string for string-list:	
sss	
Enter string for string-list:	
ddd	
Enter string for string-list:	
fff	
{10: 'sss', 11: 'ddd', 12: 'fff'}	

Test Case - 2	
User Output	
Enter size of int-list:	
2	
Enter size of string-list:	
3	
Enter int for int-list:	
56	
Enter int for int-list:	
57	
Enter string for string-list:	
Ronaldo	
Enter string for string-list:	
Bekham	
Enter string for string-list:	
Christiana	
{56: 'Ronaldo', 57: 'Bekham'}	

S.No: 11

Exp. Name: **Write a Program to Count frequency of a given character in a given File**

Date: 2023-02-01

Aim:

Write a program to `count` frequency of a given character in a given file.

Can you use character frequency to tell whether the given file is a **Python** program file, **C** program file or a **text** file?

At the time of execution, the program should print the following messages one by one on the console as:

Enter the filename:

Enter the character to be searched:

For example, if the user gives the input as:

Enter the filename: Lab6b.txt

Enter the character to be searched: J

then the program should print the result as:

J appears 6 times in file

The type of file:

Note: Write the code in the file `Lab6b.py` and Let us consider the text file `Lab6b.txt` to run the program.

Source Code:

`Lab6b.py`

```
f=input("Enter the filename: ")
s=input("Enter the character to be searched: ")
file=open(f,'r')
text=file.read()
print("{} appears {} times in file".format(s,text.count(s)))
print("The type of file:",type(file))
```

`Lab6b.txt`

JNTUK – Jawaharlal Nehru Technological University, Kakinada

JNTUH – Jawaharlal Nehru Technological University, Hyderabad

JNTUA – Jawaharlal Nehru Technological University, Anantapur

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter the filename:

Lab6b.txt

Enter the character to be searched:

J

J appears 6 times in file

The type of file:

Test Case - 2

User Output

Enter the filename:

Lab6b.txt

Enter the character to be searched:

k

k appears 1 times in file

The type of file:

Test Case - 3

User Output

Enter the filename:

Lab6b.txt

Enter the character to be searched:

l

l appears 12 times in file

The type of file:

S.No: 12

Exp. Name: **Write a Program to Print Lines of a File in Reverse order**

Date: 2023-02-01

Aim:

Write a program to print lines of a file in **reverse order**, i.e. last line first.....first line last.

At the time of execution, the program should print the message on the console as:

Enter file name:

For example, if the user gives the input as:

Enter file name: Input1.txt

then the program should print the result as:

```
VTU Visvesvaraya Technological University, Belgaum  
RTU Rajasthan Technical University, Jaipur  
AnnaUniv Anna University, Chennai  
JNTUK Jawaharlal Nehru Technological University, Kakinada  
AU Andhra University, Vizag  
JNU Jawaharlal Nehru University, Delhi  
JNTUH Jawaharlal Nehru Technological University, Hyderabad
```

Note: Write the code in the file **Lab7a.py** and Let us consider the text files **Input1.txt**, **Input2.txt**, **Input3.txt** and **Input4.txt** to run the program.

Source Code:

Lab7a.py

```
fn=input("Enter file name: ")  
file=open(fn,'r')  
lst=file.readlines()  
lst2=lst[::-1]  
for i in lst2:  
    i=i.replace("\n","")  
    print(i)
```

Input1.txt

```
JNTUH Jawaharlal Nehru Technological University, Hyderabad  
JNU Jawaharlal Nehru University, Delhi  
AU Andhra University, Vizag  
JNTUK Jawaharlal Nehru Technological University, Kakinada  
AnnaUniv Anna University, Chennai  
RTU Rajasthan Technical University, Jaipur  
VTU Visvesvaraya Technological University, Belgaum
```

Input2.txt

A. P. J. Abdul Kalam – Quotes

Man needs his difficulties because they are necessary to enjoy success.
You have to dream before your dreams can come true.
We should not give up and we should not allow the problem to defeat us.
Great dreams of great dreamers are always transcended.
Look at the sky. We are not alone. The whole universe is friendly to us and
conspires only to give the best to those who dream and work.
Do we not realize that self respect comes with self reliance?
Be more dedicated to making solid achievements than in running after swift but
synthetic happiness.
Unless India stands up to the world, no one will respect us. In this world, fear has
no place. Only strength respects strength.
Climbing to the top demands strength, whether it is to the top of Mount Everest or
to the top of your career.
Those who cannot work with their hearts achieve but a hollow, half-hearted success
that breeds bitterness all around.

Input3.txt

Swami Vivekananda – Quotes

Arise, awake and do not stop until the goal is reached.
You have to grow from the inside out. None can teach you, none can make you
spiritual. There is no other teacher but your own soul.
You cannot believe in God until you believe in yourself.
Condemn none: if you can stretch out a helping hand, do so. If you cannot, fold your
hands, bless your brothers, and let them go their own way.
We are what our thoughts have made us; so take care about what you think. Words are
secondary. Thoughts live; they travel far.
All the powers in the universe are already ours. It is we who have put our hands
before our eyes and cry that it is dark.
The world is the great gymnasium where we come to make ourselves strong.
Where can we go to find God if we cannot see Him in our own hearts and in every
living being.
All differences in this world are of degree, and not of kind, because oneness is the
secret of everything.
When an idea exclusively occupies the mind, it is transformed into an actual
physical or mental state.

Input4.txt

Ramana Maharshi – Quotes

Your own Self-realization is the greatest service you can render the world.
No one succeeds without effort... Those who succeed owe their success to
perseverance.
The degree of freedom from unwanted thoughts and the degree of concentration on a
single thought are the measures to gauge spiritual progress.

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter file name:
Input1.txt
VTU Visvesvaraya Technological University, Belgaum
RTU Rajasthan Technical University, Jaipur
AnnaUniv Anna University, Chennai
JNTUK Jawaharlal Nehru Technological University, Kakinada
AU Andhra University, Vizag
JNU Jawaharlal Nehru University, Delhi
JNTUH Jawaharlal Nehru Technological University, Hyderabad

Test Case - 2
User Output
Enter file name:
Input2.txt
Those who cannot work with their hearts achieve but a hollow, half-hearted success that breeds bitterness all around.
Climbing to the top demands strength, whether it is to the top of Mount Everest or to the top of your career.
Unless India stands up to the world, no one will respect us. In this world, fear has no place. Only strength respects strength.
Be more dedicated to making solid achievements than in running after swift but synthetic happiness.
Do we not realize that self respect comes with self reliance?
Look at the sky. We are not alone. The whole universe is friendly to us and conspires only to give the best to those who dream and work.
Great dreams of great dreamers are always transcended.
We should not give up and we should not allow the problem to defeat us.
You have to dream before your dreams can come true.
Man needs his difficulties because they are necessary to enjoy success.
A. P. J. Abdul Kalam – Quotes

Test Case - 3
User Output
Enter file name:
Input4.txt
The degree of freedom from unwanted thoughts and the degree of concentration on a single thought are the measures to gauge spiritual progress.
No one succeeds without effort... Those who succeed owe their success to perseverance.
Your own Self-realization is the greatest service you can render the world.
Ramana Maharshi – Quotes

S.No: 13

Exp. Name: **Write a Program to compute the number of Characters, Words and Lines in a File**

Date: 2023-02-01

Aim:

Write a program to compute the number of **characters**, **words** and **lines** in a file.

At the time of execution, the program should print the message on the console as:

Enter file name:

For example, if the user gives the input as:

Enter file name: Input1.txt

then the program should print the result as:

Lines = 7
words = 35
Characters = 311

Note: Write the code in the file [Lab7b.py](#) and Let us consider the text files [Input1.txt](#), [Input2.txt](#), [Input3.txt](#) and [Input4.txt](#) to run the program.

Source Code:

[Lab7b.py](#)

```
fn=input("Enter file name: ")  
file=open(fn,'r')  
l=0  
w=0  
c=0  
for i in file:  
    i=i.strip()  
    words=i.split()  
    l+=1  
    w+=len(words)  
    c+=len(i)  
print("Lines =",l)  
print("words =",w)  
print("Characters =",c+(l-1))
```

[Input1.txt](#)

JNTUH Jawaharlal Nehru Technological University, Hyderabad
JNU Jawaharlal Nehru University, Delhi
AU Andhra University, Vizag
JNTUK Jawaharlal Nehru Technological University, Kakinada
AnnaUniv Anna University, Chennai
RTU Rajasthan Technical University, Jaipur
VTU Visvesvaraya Technological University, Belgaum

[Input2.txt](#)

A. P. J. Abdul Kalam – Quotes

Man needs his difficulties because they are necessary to enjoy success.
You have to dream before your dreams can come true.
We should not give up and we should not allow the problem to defeat us.
Great dreams of great dreamers are always transcended.
Look at the sky. We are not alone. The whole universe is friendly to us and
conspires only to give the best to those who dream and work.
Do we not realize that self respect comes with self reliance?
Be more dedicated to making solid achievements than in running after swift but
synthetic happiness.
Unless India stands up to the world, no one will respect us. In this world, fear has
no place. Only strength respects strength.
Climbing to the top demands strength, whether it is to the top of Mount Everest or
to the top of your career.
Those who cannot work with their hearts achieve but a hollow, half-hearted success
that breeds bitterness all around.

Input3.txt

Swami Vivekananda – Quotes

Arise, awake and do not stop until the goal is reached.
You have to grow from the inside out. None can teach you, none can make you
spiritual. There is no other teacher but your own soul.
You cannot believe in God until you believe in yourself.
Condemn none: if you can stretch out a helping hand, do so. If you cannot, fold your
hands, bless your brothers, and let them go their own way.
We are what our thoughts have made us; so take care about what you think. Words are
secondary. Thoughts live; they travel far.
All the powers in the universe are already ours. It is we who have put our hands
before our eyes and cry that it is dark.
The world is the great gymnasium where we come to make ourselves strong.
Where can we go to find God if we cannot see Him in our own hearts and in every
living being.
All differences in this world are of degree, and not of kind, because oneness is the
secret of everything.
When an idea exclusively occupies the mind, it is transformed into an actual
physical or mental state.

Input4.txt

Ramana Maharshi – Quotes

Your own Self-realization is the greatest service you can render the world.
No one succeeds without effort... Those who succeed owe their success to
perseverance.
The degree of freedom from unwanted thoughts and the degree of concentration on a
single thought are the measures to gauge spiritual progress.

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter file name:

Input1.txt

Lines = 7

words = 35

Characters = 311

Test Case - 2

User Output

Enter file name:

Input2.txt

Lines = 11

words = 168

Characters = 935

S.No: 14

Exp. Name: **Write a Function ball_collide that takes Two Balls as parameters and Computes if they are Colliding**

Date: 2023-02-01

Aim:

Write a function **ball_collide()** that takes two balls as parameters and computes if they are colliding.

The function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r) where **x**, **y** are the coordinates of the centre and **r** being the radius.

If (distance between two balls centers) \leq (sum of their radius) then they are colliding.

The formula's are:

- **Distance between the centres =** $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- **Sum of their radius =** $r_1 + r_2$

We use the exponentiation operator ****** to calculate square root and square.

At the time of execution, the program should print the following messages one by one on the console as:

Enter x, y, r of ball-1:
Enter x, y, r of ball-2:

For example, if the user gives the input as:

Enter x, y, r of ball-1: 0 0 3
Enter x, y, r of ball-2: 3 3 3

then the program should print the result as:

True – Balls are colliding

Enter x, y, r of ball-1:
Enter x, y, r of ball-2:

For example, if the user gives the input as:

Enter x, y, r of ball-1: 0 0 1
Enter x, y, r of ball-2: 3 3 1

then the program should print the result as:

False – Balls are not colliding

Source Code:

Lab8a.py

```

import math
def ballcollided():
    s1=input("Enter x, y, r of ball-1: ")
    x1,y1,r1=s1.split(" ")
    x1=int(x1)
    y1=int(y1)
    r1=int(r1)
    s2=input("Enter x, y, r of ball-2: ")
    x2,y2,r2=s2.split(" ")
    x2=int(x2)
    y2=int(y2)
    r2=int(r2)
    d=math.sqrt((x2-x1)**2+(y2-y1)**2)
    s=r1+r2
    if d<=s:
        print("True - Balls are colliding")
    else:
        print("False - Balls are not colliding")
ballcollided()

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter x, y, r of ball-1:
113
Enter x, y, r of ball-2:
113
True - Balls are colliding

Test Case - 2

User Output

Enter x, y, r of ball-1:
0 0 3
Enter x, y, r of ball-2:
3 3 3
True - Balls are colliding

Test Case - 3

User Output

Enter x, y, r of ball-1:
0 0 1
Enter x, y, r of ball-2:
3 3 1
False - Balls are not colliding

S.No: 15

Exp. Name: **Write a Program to find Mean, Median, Mode, Standard deviation, Variance for the given set of numbers in a List**

Date: 2023-02-01

Aim:

Write a program to find mean, median, mode, standard deviation and variance for the given set of numbers in a list using statistics.

- **Mean:** Return the sample arithmetic mean (average) of data.
- **Median:** Return the median (middle value) of numeric data.
 - When the number of data points is odd, return the middle data point.
 - When the number of data points is even, the median is interpolated by taking the average of the two middle values.
- **Mode:** Return the most common data (most repeated data) point from discrete or nominal data.
- **Standard deviation:** Quantify the amount of variation or dispersion of a set of data values.
- **Variance:** Expectation of the squared deviation of a random variable from its mean.

At the time of execution, the program should print the message on the console as:

Enter size of list:

For example, if the user gives the **input** as:

Enter size of list: 4

Now, the program should print the following messages one by one on the console as:

Enter integer value:
Enter integer value:
Enter integer value:
Enter integer value:

For example, if the user gives the **input** as:

Enter integer value: 2
Enter integer value: 3
Enter integer value: 4
Enter integer value: 3

then the program should **print** the result as:

Mean of list: 3
Median of list: 3.0
Mode of list: 3
Standard deviation of list: 0.816496580927726
Variance of list: 0.6666666666666666

Source Code:

Lab8b.py

```

import statistics
import statistics as st
n=int(input("Enter size of list: "))
l=[]
for i in range(n):
    e=int(input("Enter integer value: "))
    l.append(e)
print("Mean of list:",st.mean(l))
print("Median of list:",st.median(l))
print("Mode of list:",st.mode(l))
print("Standard deviation of list:",st.stdev(l))
print("Variance of list:",st.variance(l))

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter size of list:
4
Enter integer value:
2
Enter integer value:
3
Enter integer value:
4
Enter integer value:
3
Mean of list: 3
Median of list: 3.0
Mode of list: 3
Standard deviation of list: 0.816496580927726
Variance of list: 0.6666666666666666

S.No: 16

Exp. Name: **Write a Function nearly_equal() to test whether Two Strings are nearly Equal.**

Date: 2023-02-01

Aim:

Write a function **nearly_equal()** to test whether two **strings** are nearly equal. Two strings **str1** and **str2** are nearly equal when **str1** can be generated by a single mutation on **str2**.

Source Code:

Lab9a.py

```
def neralyequal():
    s1=input("Please enter a string: ")
    s2=input("Please enter another string: ")
    l1=len(s1)
    l2=len(s2)
    if(l1>l2 and (l1-l2)==1):
        print("Some Character is extra in string1")
        r=s1[1]
        s1=s1.replace(r,"")
        print("Strings after removal of the extra character:",s1,s2)
        print("False")
    elif(l2>l1 and (l2-l1)==1):
        print("Some Character is extra in string1")
        r=s2[1]
        s2=s2.replace(r,"")
        print("Strings after removal of the extra character:",s2,s1)
        print("False")
    elif(l1==l2):
        print("Strings are of equal Length")
        print("True")
    else:
        print("Lengths differ by more than 1")
        print("False")
neralyequal()
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Please enter a string:

James

Please enter another string:

James

Strings are of equal Length

True

Test Case - 2

User Output

Please enter a string:

James
Please enter another string:
John
Some Character is extra in string1
Strings after removal of the extra character: Jmes John
False

Test Case - 3
User Output
Please enter a string:
CodeTantra
Please enter another string:
CodeVita
Lengths differ by more than 1
False

S.No: 17

Exp. Name: ***Write a function dups to find all duplicates in the list***

Date: 2023-02-01

Aim:

Write a function **dups** to find all duplicates in the list.

At the time of execution, the program should print the message on the console as:

Enter size of list:

For example, if the user gives the input as:

Enter size of list: 5

Now, the program should print the following messages one by one on the console as:

Enter value:
Enter value:
Enter value:
Enter value:
Enter value:

For example, if the user gives the input as:

Enter value: 5
Enter value: 4
Enter value: 1
Enter value: 2
Enter value: 1

then the program should print the result as:

Original List = [5, 4, 1, 2, 1]
Duplicates = [1]

Source Code:

Lab9b.py

Execution Results - All test cases have succeeded!

Test Case - 1

```
Original List = [5, 4, 1, 2, 1]
```

```
Duplicates = [1]
```

Test Case - 2

User Output

```
Enter size of list:
```

```
6
```

```
Enter value:
```

```
12
```

```
Enter value:
```

```
52
```

```
Enter value:
```

```
96
```

```
Enter value:
```

```
12
```

```
Enter value:
```

```
56
```

```
Enter value:
```

```
96
```

```
Original List = [12, 52, 96, 12, 56, 96]
```

```
Duplicates = [12, 96]
```

S.No: 18

Exp. Name: ***Write a function unique to find all the unique elements of a list***

Date: 2023-02-01

Aim:

Write a function **unique** to find all the unique elements of a list.

At the time of execution, the program should print the message on the console as:

Enter size of list:

For example, if the user gives the input as:

Enter size of list: 6

Now, the program should print the following messages one by one on the console as:

Enter value:
Enter value:
Enter value:
Enter value:
Enter value:
Enter value:

For example, if the user gives the input as:

Enter value: 8
Enter value: 8
Enter value: 15
Enter value: 96
Enter value: 45
Enter value: 8

then the program should print the result as:

Original List = [8, 8, 15, 96, 45, 8]
Unique elements = [8, 15, 96, 45]

Source Code:

Lab9c.py

```

'''def unique(list1):

#Write your code here...

list1 = []      #List declaration
n = int(input("Enter size of list: "))
for i in range(n):
    val = int(input("Enter value: "))

#Write here.....



print("Original List =", )
print("Unique elements =", )'''
n=int(input("Enter size of list: "))
ol=[]
ul=[]
c=0
for i in range(n):
    e=int(input("Enter value: "))
    ol.append(e)
for i in ol:
    '''for j in ol:
        if i==j:
            c=c+1
    if c==1:''
        if i not in ul:
            ul.append(i)
print("Original List =",ol)
print("Unique elements =",ul)

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter size of list:
4
Enter value:
5
Enter value:
63
Enter value:
5
Enter value:
4
Original List = [5, 63, 5, 4]
Unique elements = [5, 63, 4]

Test Case - 2

User Output

Enter size of list:

5

Enter value:

12

Enter value:

96

Enter value:

3

Enter value:

12

Enter value:

3

Original List = [12, 96, 3, 12, 3]

Unique elements = [12, 96, 3]

S.No: 19

Exp. Name: **Write a function cumulative_product to compute cumulative product of a list of numbers**

Date: 2023-08-30

Aim:

Write a function **cumulative_product** to compute cumulative product of a list of numbers.

At the time of execution, the program should print the message on the console as:

Enter size of list:

For example, if the user gives the input as:

Enter size of list: 3

Now, the program should print the following messages one by one on the console as:

Enter value:
Enter value:
Enter value:

For example, if the user gives the input as:

Enter value: 2
Enter value: 5
Enter value: 3

then the program should print the result as:

Original List = [2, 5, 3]
Cumulative Product List = [2, 10, 30]

Source Code:

Lab10a.py

```
#def cumulative_product(list1):
#Write your code here...
n=int(input("Enter size of list: "))
ol=[]
cp=[]
for i in range(n):
    e=int(input("Enter value: "))
    ol.append(e)
cp.append(ol[0])
e=ol[0]
for i in range(1,len(ol)):
    e=e*ol[i]
    cp.append(e)
print("Original List =",ol)
print("Cumulative Product List =",cp)
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter size of list:

3

Enter value:

2

Enter value:

5

Enter value:

3

Original List = [2, 5, 3]

Cumulative Product List = [2, 10, 30]

Test Case - 2

User Output

Enter size of list:

6

Enter value:

5

Enter value:

6

Enter value:

2

Enter value:

1

Enter value:

3

Enter value:

2

Original List = [5, 6, 2, 1, 3, 2]

Cumulative Product List = [5, 30, 60, 60, 180, 360]

S.No: 20

Exp. Name: **Write a function reverse to reverse a list without using the reverse function**

Date: 2023-08-30

Aim:

Write a function **reverse** to reverse a list without using the reverse function.

At the time of execution, the program should print the message on the console as:

Enter size of list:

For example, if the user gives the input as:

Enter size of list: 3

Now, the program should print the following messages one by one on the console as:

Enter value:

Enter value:

Enter value:

For example, if the user gives the input as:

Enter value: 66

Enter value: 33

Enter value: 56

then the program should print the result as:

Original List = [66, 33, 56]

Reversed List = [56, 33, 66]

Source Code:

Lab10b.py

```
n=int(input("Enter size of list: "))
ol=[]
for i in range(n):
    e=int(input("Enter value: "))
    ol.append(e)
print("Original List =",ol)
print("Reversed List =",ol[::-1])
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter size of list:

6

Enter value:

1

Enter value:

3
Enter value:
5
Enter value:
6
Enter value:
2
Enter value:
8
Original List = [1, 3, 5, 6, 2, 8]
Reversed List = [8, 2, 6, 5, 3, 1]

Test Case - 2
User Output
Enter size of list:
4
Enter value:
95
Enter value:
63
Enter value:
45
Enter value:
12
Original List = [95, 63, 45, 12]
Reversed List = [12, 45, 63, 95]

S.No: 21

Exp. Name: **Write function to compute gcd, lcm of two numbers**

Date: 2023-08-30

Aim:

Write function to compute **gcd, lcm** of two numbers.

Each function shouldn't exceed one line.

GCD - Greatest Common divisor - maximum value which can divide both numbers.

LCM - Lowest Common Multiple - is the minimum value which is divisible by both numbers.

At the time of execution, the program should print the message on the console as:

Enter an integer value:
Enter an integer value:

For example, if the user gives the input as:

Enter an integer value: 25
Enter an integer value: 55

then the program should print the result as:

GCD of 25 and 55 is 5
LCM of 25 and 55 is 275

Source Code:

Lab10c.py

```
import math
a=int(input("Enter an integer value: "))
b=int(input("Enter an integer value: "))
print("GCD of {} and {} is {}".format(a,b,math.gcd(a,b)))
print("LCM of {} and {} is {}".format(a,b,abs(a*b)//math.gcd(a,b)))
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter an integer value:

25

Enter an integer value:

16

GCD of 25 and 16 is 1

LCM of 25 and 16 is 400

Test Case - 2

User Output

Enter an integer value:

10

Enter an integer value:

10

GCD of 10 and 10 is 10

LCM of 10 and 10 is 10

S.No: 22

Exp. Name: **Write a Python program that defines a matrix and prints Matrices**

Date: 2023-08-30

Aim:

Write a **Python** program that defines a matrix and prints matrices.

A matrix is a rectangular table of elements. The following is a 2x4 matrix, meaning there are 2 rows, 4 columns:

Matrix Representation:

5	4	7	11
3	3	8	17

In **Python** and other programming languages, a matrix is often represented with a list of lists. The sample matrix above could be created with:

matrix = [[5, 4, 7, 11], [3, 3, 8, 17]] So **matrix[0][0]** is 5, **matrix[0][1]** is 4, **matrix[1][0]** is 3 and so on.

Sample Input and Output:

```
Number of rows, m = 2
Number of columns, n = 2
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 2 column: 1
Entry in row: 2 column: 2
[[1, 2], [3, 4]]
```

Source Code:

Lab11a.py

```
R=int(input("Number of rows, m = "))
C=int(input("Number of columns, n = "))
matrix=[]
for i in range(R):
    a=[]
    for j in range(C):
        print("Entry in row: {} column: {}".format(i+1,j+1))
        a.append(int(input()))
    matrix.append(a)
print(matrix)
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Number of rows, m =

2

Number of columns, n =

2
Entry in row: 1 column: 1
1
Entry in row: 1 column: 2
2
Entry in row: 2 column: 1
3
Entry in row: 2 column: 2
4
[[1, 2], [3, 4]]

Test Case - 2	
User Output	
Number of rows, m =	
3	
Number of columns, n =	
2	
Entry in row: 1 column: 1	
1	
Entry in row: 1 column: 2	
2	
Entry in row: 2 column: 1	
3	
Entry in row: 2 column: 2	
4	
Entry in row: 3 column: 1	
5	
Entry in row: 3 column: 2	
6	
[[1, 2], [3, 4], [5, 6]]	

Test Case - 3	
User Output	
Number of rows, m =	
3	
Number of columns, n =	
3	
Entry in row: 1 column: 1	
11	
Entry in row: 1 column: 2	
22	
Entry in row: 1 column: 3	
33	
Entry in row: 2 column: 1	
44	
Entry in row: 2 column: 2	
55	
Entry in row: 2 column: 3	

66

Entry in row: 3 column: 1

77

Entry in row: 3 column: 2

88

Entry in row: 3 column: 3

99

[[11, 22, 33], [44, 55, 66], [77, 88, 99]]

Test Case - 4

User Output

Number of rows, m =

2

Number of columns, n =

2

Entry in row: 1 column: 1

-11

Entry in row: 1 column: 2

-23

Entry in row: 2 column: 1

-63

Entry in row: 2 column: 2

-78

[[-11, -23], [-63, -78]]

Test Case - 5

User Output

Number of rows, m =

3

Number of columns, n =

4

Entry in row: 1 column: 1

12

Entry in row: 1 column: 2

13

Entry in row: 1 column: 3

14

Entry in row: 1 column: 4

15

Entry in row: 2 column: 1

16

Entry in row: 2 column: 2

17

Entry in row: 2 column: 3

18

Entry in row: 2 column: 4

19

Entry in row: 3 column: 1

20

Entry in row: 3 column: 2

21

Entry in row: 3 column: 3

22

Entry in row: 3 column: 4

23

`[[12, 13, 14, 15], [16, 17, 18, 19], [20, 21, 22, 23]]`

S.No: 23

Exp. Name: ***Write a Python program to perform addition of two matrices***

Date: 2023-08-30

Aim:

Write a **Python** program to find addition of two matrices.

Sample Input and Output-1:

```
Number of rows for matrix - A, m = 2
Number of columns for matrix - A, n = 3
Number of rows for matrix - B, p = 2
Number of columns for matrix - B, q = 3
Enter values for matrix - A
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 1 column: 3
Entry in row: 2 column: 1
Entry in row: 2 column: 2
Entry in row: 2 column: 3
Enter values for matrix - B
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 1 column: 3
Entry in row: 2 column: 1
Entry in row: 2 column: 2
Entry in row: 2 column: 3
Matrix a = [[11, 22, 33], [44, 55, 66]]
Matrix b = [[1, 2, 3], [4, 5, 6]]
Addition of two matrices = [[12, 24, 36], [48, 60, 72]]
```

Sample Input and Output-2:

```
Number of rows for matrix - A, m = 2
Number of columns for matrix - A, n = 2
Number of rows for matrix - B, p = 2
Number of columns for matrix - B, q = 3
Addition is not possible
```

Source Code:

Lab11b.py

```

R1=int(input("Number of rows for matrix - A, m = "))
C1=int(input("Number of columns for matrix - A, n = "))
R2=int(input("Number of rows for matrix - B, p = "))
C2=int(input("Number of columns for matrix - B, q = "))
matrix1=[]
matrix2=[]
if R1==R2 and C1==C2:
    print("Enter values for matrix - A")
    for i in range(R1):
        a=[]
        for j in range(C1):
            print("Entry in row: {} column: {}".format(i+1,j+1))
            a.append(int(input()))
        matrix1.append(a)
    print("Enter values for matrix - B")
    for i in range(R2):
        a=[]
        for j in range(C2):
            print("Entry in row: {} column: {}".format(i+1,j+1))
            a.append(int(input()))
        matrix2.append(a)
    print("Matrix a =",matrix1)
    print("Matrix b =",matrix2)
    matrixsum=[]
    for i in range(len(matrix1)):
        matrixsum.append([])
        for j in range(len(matrix1[0])):
            matrixsum[i].append(matrix1[i][j]+matrix2[i][j])
    print("Addition of two matrices =",matrixsum)
else:
    print("Addition is not possible")

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Number of rows for matrix - A, m =

2

Number of columns for matrix - A, n =

3

Number of rows for matrix - B, p =

2

Number of columns for matrix - B, q =

3

Enter values for matrix - A

Entry in row: 1 column: 1

11

Entry in row: 1 column: 2

22

Entry in row: 1 column: 3

33

Entry in row: 2 column: 1

44

Entry in row: 2 column: 2

55

Entry in row: 2 column: 3

66

Enter values for matrix - B

Entry in row: 1 column: 1

1

Entry in row: 1 column: 2

2

Entry in row: 1 column: 3

3

Entry in row: 2 column: 1

4

Entry in row: 2 column: 2

5

Entry in row: 2 column: 3

6

Matrix a = [[11, 22, 33], [44, 55, 66]]

Matrix b = [[1, 2, 3], [4, 5, 6]]

Addition of two matrices = [[12, 24, 36], [48, 60, 72]]

Test Case - 2

User Output

Number of rows for matrix - A, m =

2

Number of columns for matrix - A, n =

2

Number of rows for matrix - B, p =

2

Number of columns for matrix - B, q =

3

Addition is not possible

Test Case - 3

User Output

Number of rows for matrix - A, m =

2

Number of columns for matrix - A, n =

2

Number of rows for matrix - B, p =

2

Number of columns for matrix - B, q =

2

Enter values for matrix - A

Entry in row: 1 column: 1

1

```
Entry in row: 1 column: 2
2
Entry in row: 2 column: 1
3
Entry in row: 2 column: 2
4
Enter values for matrix - B
Entry in row: 1 column: 1
1
Entry in row: 1 column: 2
2
Entry in row: 2 column: 1
3
Entry in row: 2 column: 2
4
Matrix a = [[1, 2], [3, 4]]
Matrix b = [[1, 2], [3, 4]]
Addition of two matrices = [[2, 4], [6, 8]]
```

Test Case - 4

User Output

```
Number of rows for matrix - A, m =
3
Number of columns for matrix - A, n =
3
Number of rows for matrix - B, p =
3
Number of columns for matrix - B, q =
3
Enter values for matrix - A
Entry in row: 1 column: 1
1
Entry in row: 1 column: 2
2
Entry in row: 1 column: 3
3
Entry in row: 2 column: 1
4
Entry in row: 2 column: 2
5
Entry in row: 2 column: 3
6
Entry in row: 3 column: 1
7
Entry in row: 3 column: 2
8
Entry in row: 3 column: 3
9
Enter values for matrix - B
Entry in row: 1 column: 1
```

9
Entry in row: 1 column: 2
8
Entry in row: 1 column: 3
7
Entry in row: 2 column: 1
6
Entry in row: 2 column: 2
5
Entry in row: 2 column: 3
4
Entry in row: 3 column: 1
3
Entry in row: 3 column: 2
2
Entry in row: 3 column: 3
1
Matrix a = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Matrix b = [[9, 8, 7], [6, 5, 4], [3, 2, 1]]
Addition of two matrices = [[10, 10, 10], [10, 10, 10], [10, 10, 10]]

S.No: 24

Exp. Name: ***Write a Python program to perform multiplication of two matrices***

Date: 2023-08-30

Aim:

Write a Python program to perform **multiplication** of two matrices.

Sample Input and Output-1:

```
Enter values for matrix - A
Number of rows, m = 2
Number of columns, n = 2
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 2 column: 1
Entry in row: 2 column: 2
Enter values for matrix - B
Number of rows, m = 2
Number of columns, n = 2
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 2 column: 1
Entry in row: 2 column: 2
Matrix - A = [[1, 2], [3, 4]]
Matrix - B = [[1, 2], [3, 4]]
Matrix - A * Matrix- B = [[7, 10], [15, 22]]
```

Sample Input and Output-2:

```
Enter values for matrix - A
Number of rows, m = 2
Number of columns, n = 3
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 1 column: 3
Entry in row: 2 column: 1
Entry in row: 2 column: 2
Entry in row: 2 column: 3
Enter values for matrix - B
Number of rows, m = 2
Number of columns, n = 3
Entry in row: 1 column: 1
Entry in row: 1 column: 2
Entry in row: 1 column: 3
Entry in row: 2 column: 1
Entry in row: 2 column: 2
Entry in row: 2 column: 3
Matrix - A = [[1, 2, 3], [4, 5, 6]]
Matrix - B = [[1, 2, 3], [4, 5, 6]]
Cannot multiply the two matrices. Incorrect dimensions.
Matrix - A * Matrix- B = None
```

Source Code:

Lab11c.py

```

def matmult(A, B,r1,c1,r2,c2):
    if c1==r2:
        if(c1==c2):
            la=len(A)
            lao=len(A[0])
            lb=c1
        else:
            la=len(A)
            lao=len(B[0])
            lb=r2
    result=[]
    r=0
    for i in range(la):
        tr=[]
        for j in range(lao):
            for k in range(lb):
                s=(A[i][k]*B[k][j])
                r=r+s
            tr.append(r)
            r=0
        result.append(tr)
        del tr
    print("Matrix - A * Matrix- B =",result)
else:
    print("Cannot multiply the two matrices. Incorrect dimensions.")
    print("Matrix - A * Matrix- B = None")
def readmatrix(name = ''):
    print("Enter values for",name)
    R=int(input("Number of rows, m = "))
    C=int(input("Number of columns, n = "))
    mat=[]
    for i in range(R):
        a=[]
        for j in range(C):
            print("Entry in row: {} column: {}".format(i+1,j+1))
            a.append(int(input()))
        mat.append(a)
    return mat,R,C
matrixa,ra,ca = readmatrix('matrix - A')
matrixb,rb,cb = readmatrix('matrix - B')
print("Matrix - A =", matrixa)
print("Matrix - B =", matrixb)
matmult(matrixa, matrixb,ra,ca,rb,cb)

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter values for matrix - A

Number of rows, m =

2

Number of columns, n =

2
Entry in row: 1 column: 1
1
Entry in row: 1 column: 2
2
Entry in row: 2 column: 1
3
Entry in row: 2 column: 2
4
Enter values for matrix - B
Number of rows, m =
2
Number of columns, n =
2
Entry in row: 1 column: 1
1
Entry in row: 1 column: 2
2
Entry in row: 2 column: 1
3
Entry in row: 2 column: 2
4
Matrix - A = [[1, 2], [3, 4]]
Matrix - B = [[1, 2], [3, 4]]
Matrix - A * Matrix- B = [[7, 10], [15, 22]]

Test Case - 2
User Output
Enter values for matrix - A
Number of rows, m =
2
Number of columns, n =
3
Entry in row: 1 column: 1
1
Entry in row: 1 column: 2
2
Entry in row: 1 column: 3
3
Entry in row: 2 column: 1
4
Entry in row: 2 column: 2
5
Entry in row: 2 column: 3
6
Enter values for matrix - B
Number of rows, m =
3
Number of columns, n =
2

Entry in row: 1 column: 1

1

Entry in row: 1 column: 2

2

Entry in row: 2 column: 1

3

Entry in row: 2 column: 2

4

Entry in row: 3 column: 1

5

Entry in row: 3 column: 2

6

Matrix - A = [[1, 2, 3], [4, 5, 6]]

Matrix - B = [[1, 2], [3, 4], [5, 6]]

Matrix - A * Matrix- B = [[22, 28], [49, 64]]

Test Case - 3

User Output

Enter values for matrix - A

Number of rows, m =

3

Number of columns, n =

2

Entry in row: 1 column: 1

1

Entry in row: 1 column: 2

2

Entry in row: 2 column: 1

3

Entry in row: 2 column: 2

3

Entry in row: 3 column: 1

2

Entry in row: 3 column: 2

1

Enter values for matrix - B

Number of rows, m =

2

Number of columns, n =

1

Entry in row: 1 column: 1

1

Entry in row: 2 column: 1

2

Matrix - A = [[1, 2], [3, 3], [2, 1]]

Matrix - B = [[1], [2]]

Matrix - A * Matrix- B = [[5], [9], [4]]