

1. Write a python program to check whether the given number is in a range.

```
lower = int(input("Enter the lower range: "))
upper = int(input("Enter the upper range: "))
if lower <= upper:
    num = int(input("Enter a number to check: "))
    if (num >= lower) and (num <= upper):
        print("Number is in the given range.")
    else:
        print("Number is not in the range.")
else:
    print("Lower bound should be less than or equal to the upper bound!!!")
```

```
== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p01.py =
Enter the lower range: 34
Enter the upper range: 45
Enter a number to check: 43
Number is in the given range.

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p01.py ==
Enter the lower range: 44
Enter the upper range: 22
Lower bound should be less than or equal to the upper bound!!!
```

2. Write a python program to calculate the distance between two points taking input from the user using the Pythagorean theorem.

```
import math;
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: "))
dist = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
print("Distance between these two points: ", dist)
```

```

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p02.py ==
Enter the x-coordinate of first point: 2
Enter the y-coordinate of first point: 2
Enter the x-coordinate of second point: 5
Enter the y-coordinate of second point: 6
Distance between these two points: 5.0

```

3. A library charges a fine for every book returned late. For the first 5 days, the fine is 50 paise, for 6-10 days fine is one rupee and above 10 days fine is 5 rupees. If you return the book after 30 days your membership will be cancelled. Write a Python program to accept the number of days the member is late to return the book and display the fine or the appropriate message.

```
days = int(input("Enter the no: of days: "))
```

```
if days <= 0:
```

```
    print('No Fine...')
```

```
elif days > 0 and days <= 5:
```

```
    print('Fine: ', 0.5*days)
```

```
elif days >=6 and days <=10:
```

```
    print('Fine: ', 1*days)
```

```
elif days >=10 and days <=30:
```

```
    print('Fine: ', 5*days)
```

```
else:
```

```
    print('Fine: ', 5*days)
```

```
    print('Membership cancelled!!!')
```

```

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p03.py ==
Enter the no: of days: -4
No Fine...

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p03.py ==
Enter the no: of days: 3
Fine:  1.5

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p03.py ==
Enter the no: of days: 8
Fine:  8

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p03.py ==
Enter the no: of days: 20
Fine:  100

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p03.py ==
Enter the no: of days: 40
Fine:  200
Membership cancelled!!!

```

4. Write a Python program to print the pattern.

```
*      *      *      *      *
*      *      *      *
*      *      *
*      *
*
```

```
rows = int(input('Enter the no: of rows: '))
```

```
for i in range (0, rows):
```

```
    for j in range (0, rows-i):
```

```
        print('*', end = ' ')
```

```
    print()
```

```
=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p04.py ==
Enter the no: of rows: 6
* * * * *
* * * *
* * *
* *
*
*
```

5. Write a Python program to print 'n' terms of the Fibonacci series of numbers.

```
n = int(input("How many terms? "))
```

```
if n <= 0:
```

```
    print("Kindly enter a positive number!!!")
```

```
elif n == 1:
```

```
    print(0)
```

```
else:
```

```
    n1 = 0
```

```
    n2 = 1
```

```
    print('Fibonacci series upto ', n)
```

```
    count = 0
```

```
    while count < n:
```

```
        print(n1, end = ' ')
```

```
        n3 = n1 + n2
```

```

n1 = n2

n2 = n3

count += 1

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p05.py ==
How many terms? -3
Kindly enter a positive number!!!

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p05.py ==
How many terms? 1
0

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p05.py ==
How many terms? 5
Fibonacci series upto 5
0 1 1 2 3

```

6. Write a Python program to find all prime numbers within a given range.

```

limit = int(input('Enter the limit: '))

print('Prime numbers:', end=' ')

for i in range(2, limit + 1, 1):

    count = 0

    for j in range (1, i + 1, 1):

        if i % j == 0:

            count += 1

    if count == 2:

        print(i, end = ' ')

```

```

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p06.py ==
Enter the limit: 20
Prime numbers: 2 3 5 7 11 13 17 19

```

7. Write a Python program to print date, time for today and now.

```

import datetime

print('Today: ', datetime.datetime.today())

print('Now: ', datetime.datetime.now())

```

```

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p07.py ==
Today:  2023-03-16 21:37:28.201748
Now:    2023-03-16 21:37:28.216474

```

8. Write a Python program to create a list and perform the following methods:  
insert(), remove(), append(), len(), pop(), clear()

```
fruits = []
n = int(input("Enter number of elements : "))
print("Enter elements")
for i in range(0, n):
    element = input()
    fruits.append(element)
print(fruits)
key = input("Enter element to append: ")
fruits.append(key)
print(fruits)
key = input("Enter element to insert: ")
pos = int(input("Enter the position: "))
fruits.insert(pos, key)
print(fruits)
key = input("Enter element to remove: ")
fruits.remove(key)
print(fruits)
print("No: of elements in the list: ", end = " ")
print(len(fruits))
pos = int(input("Enter the position of element to be popped: "))
key = fruits.pop(pos)
print("Element popped: ", key)
print(fruits)
print('Clearing the list...')
fruits.clear()
print(fruits)
```

```

=== RESTART: C:\Users\nidhi\AppData\Local\Programs\Python\Python311\Pgms\p08.py ==
Enter number of elements : 5
Enter elements
apple
banana
cherry
grape
mango
['apple', 'banana', 'cherry', 'grape', 'mango']
Enter element to append: peach
['apple', 'banana', 'cherry', 'grape', 'mango', 'peach']
Enter element to insert: orange
Enter the position: 5
['apple', 'banana', 'cherry', 'grape', 'mango', 'orange', 'peach']
Enter element to remove: cherry
['apple', 'banana', 'grape', 'mango', 'orange', 'peach']
No: of elements in the list: 6
Enter the position of element to be popped: -2
Element popped: orange
['apple', 'banana', 'grape', 'mango', 'peach']
Clearing the list...
[]

```

9. Write a Python program to create a menu to perform following operations:

- a. Addition
- b. Subtraction
- c. Multiplication
- d. Division
- e. Modulo Division
- f. Integer Division

```

def add(n1, n2):
    return n1+n2
def sub(n1, n2):
    return n1-n2
def prod(n1, n2):
    return n1*n2
def div(n1, n2):
    return n1/n2
def modulo(n1, n2):
    return n1%n2
def intdiv(n1, n2):
    return n1//n2
n1 = int(input("Enter first number: "))
n2 = int(input("Enter second number: "))

```

```

print("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n5.
Modulo Division\n6. Integer Division")
choice = int(input("Enter your choice: "))
while choice > 0 and choice < 7:
    if choice == 1:
        print("Sum is ", add(n1, n2))
    elif choice == 2:
        print("Difference is ", sub(n1, n2))
    elif choice == 3:
        print("Product is ", prod(n1, n2))
    elif choice == 4:
        print("Quotient is ", div(n1, n2))
    elif choice == 5:
        print("Remainder is ", modulo(n1, n2))
    else:
        print("Integer quotient is ", intdiv(n1, n2))
choice = int(input("Enter your choice: "))

```

```

=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p09.py ===
Enter first number: 29
Enter second number: 5
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulo Division
6. Integer Division
Enter your choice: 1
Sum is 34
Enter your choice: 2
Difference is 24
Enter your choice: 3
Product is 145
Enter your choice: 4
Quotient is 5.8
Enter your choice: 5
Remainder is 4
Enter your choice: 6
Integer quotient is 5
Enter your choice: 7

```

10. Write a Python program to find the factorial of a number using a function.

```
def factorial(n):  
    fact = 1  
    for i in range(1, n+1):  
        fact *= i  
    print("Factorial of ", n, " is ", fact)  
num = int(input("Enter a number: "))  
if num < 0:  
    print("Please enter a positive integer!!!")  
else:  
    factorial(num)
```

```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p10.py ==  
Enter a number: 8  
Factorial of 8 is 40320  
  
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p10.py ==  
Enter a number: -3  
Please enter a positive integer!!!  
  
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p10.py ==  
Enter a number: 0  
Factorial of 0 is 1
```

11. Write a Python program to create a class 'vehicle' with maxspeed, and mileage as instant attributes.

```
class Vehicle():  
    def __init__(self, maxspeed, mileage):  
        self.maxspeed = maxspeed  
        self.mileage = mileage  
    def description(self):  
        print('Maxspeed = ', self.maxspeed, '\nMileage = ', self.mileage)  
v = Vehicle(180, 20)  
print('Maximum speed of v is ', v.maxspeed)  
print('Mileage of v is ', v.mileage)  
v.description()
```



```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p11.py ===
Maximum speed of v is 180
Mileage of v is 20
Maxspeed = 180
Mileage = 20
```

12. Write a Python program to demonstrate the concept of constructor and destructor.

```
class Student:
    # constructor
    def __init__(self, name):
        print('Inside Constructor')
        self.name = name
        print('Object initialized')
    def show(self):
        print('Hello, my name is ', self.name)
    # destructor
    def __del__(self):
        print('Inside destructor')
        print('Object destroyed')

# create object
s1 = Student('Minnu')
s1.show()

# delete object
del s1
```

```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p12.py ===
Inside Constructor
Object initialized
Hello, my name is Minnu
Inside destructor
Object destroyed
```

13. Write a Python program to read the contents of the file 'mark.txt' and calculate the total marks and percentage obtained by a student.

```

f1=open("Mark.txt", "r")
n = int(f1.readline())
print('Total no: of students = ',n)
for i in range(n):
    print('Student #',i+1,':',end=' ')
    marks=(f1.readline().split())
    print(marks)
    total=0
    for j in range(len(marks)):
        total = total + int(marks[j])
    perc = float(total/500*100)
    print('Total = ',total,'\nPercentage = ',perc,'\n\n')

```

Mark.txt

4

60 70 80 90 100

55 65 75 85 60

70 60 80 90 67

89 76 56 43 90

```

=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p13.py ===
Total no: of students = 4
Student # 1 : ['60', '70', '80', '90', '100']
Total = 400
Percentage = 80.0

Student # 2 : ['55', '65', '75', '85', '60']
Total = 340
Percentage = 68.0

Student # 3 : ['70', '60', '80', '90', '67']
Total = 367
Percentage = 73.4

Student # 4 : ['89', '76', '56', '43', '90']
Total = 354
Percentage = 70.8

```

14. Write a Python program with the function 'largest' which accepts a file name as a parameter and reports the longest line in the file.

```
def findLarge(f):
    #f = open("Line.txt", "r")
    long = " "
    L = 0
    count = 0
    for line in f:
        count += 1
        print('Line - ', count)
        print(line)
        print('No: of characters = ', len(line))
        if len(line) > len(long):
            long = line
    print('Longest line: ', long, 'with characters = ', len(long))
f = open("Line.txt", "r")
findLarge(f)
```

Line.txt

MCACT305

Python Programming for Data Science

Lab programs

```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p14.py ===
Line - 1
MCACT305

No: of characters = 9
Line - 2
Python Programming for Data Science

No: of characters = 36
Line - 3
Lab programs
No: of characters = 12
Longest line: Python Programming for Data Science
with characters = 36
```

15. Write a Python program to demonstrate the exception handling for zero division error.

```
def divide(x, y):  
    try:  
        result = x / y  
    except ZeroDivisionError as e:  
        print("division by zero!, Exception raised:",e)  
    else:  
        print("result is", result)  
    finally:  
        print("executing finally clause")  
  
a = int(input("Enter value for X:"))  
b = int(input("Enter value for Y:"))  
divide(a, b)
```

```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p15.py ==  
Enter value for X:45  
Enter value for Y:5  
result is 9.0  
executing finally clause  
  
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p15.py ==  
Enter value for X:4  
Enter value for Y:0  
division by zero!, Exception raised: division by zero  
executing finally clause
```

16. Write a Python program to include multiple exceptions.

```
try:  
    x = int(input('Enter the value of x: '))  
    y = int(input('Enter the value of y: '))  
    result = x/y  
    print('Result is ', result)  
except ZeroDivisionError as e:  
    print('Division by zero is not possible!!!\nException raised: ',e)
```

```
except ValueError as v:
```

```
    print('Inappropriate value!!!\nException raised: ',v)
```

```
finally:
```

```
    print('Executing the finally clause!!!')
```

```
===== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/p16.py =====  
Enter the value of x: 4  
Enter the value of y: 3  
Result is  1.3333333333333333  
Executing the finally clause!!!
```

```
===== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/p16.py =====  
Enter the value of x: 5  
Enter the value of y: 0  
Division by zero is not possible!!!  
Exception raised:  division by zero  
Executing the finally clause!!!
```

```
===== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/p16.py =====  
Enter the value of x: 12  
Enter the value of y: as  
Inappropriate value!!!  
Exception raised:  invalid literal for int() with base 10: 'as'  
Executing the finally clause!!!
```

17. Write a Python program to create and import a calculator module name 'Calc.py'.

```
import Calc
```

```
x = int(input('Enter first number: '))
```

```
y = int(input('Enter second number: '))
```

```
a = Calc.sum(x,y)
```

```
print(f"sum={a}")
```

```
s = Calc.sub(x,y)
```

```
print(f"div={s}")
```

```
m = Calc.mul(x,y)
```

```
print(f"prod={m}")
```

```
d = Calc.div(x,y)
```

```
print(f"quo={d}")
```

```
r = Calc.rem(x,y)
```

```
print(f"rem={r}")
```

Calc.py

```
def sum(a,b):  
    return a+b  
def sub(a,b):  
    return a-b  
def mul(a,b):  
    return a*b  
def div(a,b):  
    return a/b  
def rem(a,b):  
    return a%b
```

```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p17.py ===  
Enter first number: 10  
Enter second number: 3  
sum=13  
div=7  
prod=30  
quo=3.3333333333333335  
rem=1
```

18. Write a Python program to implement database connectivity using SQLite.

```
import sqlite3  
try:  
    # Connect to DB and create a cursor  
    con = sqlite3.connect('sql.db')  
    cursor = con.cursor()  
    print('DB Init')  
    # Write a query and execute it with cursor  
    query = 'select sqlite_version();'  
    cursor.execute(query)  
    # Fetch and output result  
    result = cursor.fetchall()  
    print('SQLite Version is {}'.format(result))  
    # Close the cursor
```

```
        cursor.close()
except sqlite3.Error as error:
    print('Error occurred - ', error)
# Close DB Connection
finally:
    if con:
        con.close()
        print('SQLite Connection closed')
```

```
=== RESTART: C:/Users/nidhi/AppData/Local/Programs/Python/Python311/Pgms/p18.py ===
DB Init
SQLite Version is [('3.39.4',)]
SQLite Connection closed
```