

## Python PART B Programs

1. Write a program to perform addition, subtraction, multiplication, division, and modulo division on two integers

```
a = int(input("Enter first integer: "))
b = int(input("Enter second integer: "))
sum_result = a + b
diff_result = a - b
product_result = a * b
division_result = a / b
modulo_result = a % b
```

2. Write a program to find the greatest number from three numbers

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
max_number = max(a, b, c)
```

3. Write a program to calculate roots of a quadratic equation.

```
import math

a = float(input("Enter coefficient a: "))
b = float(input("Enter coefficient b: "))
c = float(input("Enter coefficient c: "))

discriminant = b**2 - 4*a*c

if discriminant > 0:
    root1 = (-b + math.sqrt(discriminant)) / (2*a)
    root2 = (-b - math.sqrt(discriminant)) / (2*a)
elif discriminant == 0:
    root1 = root2 = -b / (2*a)
else:
    real_part = -b / (2*a)
    imaginary_part = math.sqrt(abs(discriminant)) / (2*a)
```

4. Write a program to classify a given number as prime or composite.

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1): #This is nothing but 2 to √n for better
```

```

Time complexity.
    if num % i == 0:
        return False
    return True

num = int(input("Enter a number: "))
result = "Prime" if is_prime(num) else "Composite"

```

5. Write a program that finds whether a given character is present in a given string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

```

string = input("Enter a string: ")
char = input("Enter a character to search: ")

found = False
for i in range(len(string)):
    if string[i] == char:
        print(f"Character found at index {i}")
        found = True

if not found:
    print("Character not found in the string.")

```

6. Write a program to check whether a given year is leap year or not.

```

year = int(input("Enter a year: "))
is_leap_year = (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

```

7. Write a program to find GCD of two numbers.

```

import math

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
gcd = math.gcd(num1, num2)

```

8. Define a function which takes TWO objects representing complex numbers and returns new complex number with a addition of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition of N complex numbers

```

class Complex:
    def __init__(self, r, i):

```

```

        self.r = r
        self.i = i

    def sum(self):
        return sum(self.r), sum(self.i)

l = int(input("Enter the number of imaginary numbers: "))
r = [0] * l #Array of all the real parts

im = [0] * l #Array of all the imaginary parts
for i in range(l):
    print(f"Enter value of {i+1}. complex number(coefficients only): ")
    x = input().split(' ') #Give inputs like this -> 3 4 where 3->real part, 4-
>imaginary part
    x = [eval(i) for i in x] #We convert all those strings to numbers using eval()
    r[i], im[i] = x[0], x[1] #we add those real and imag parts to the respective
arrays

c = Complex(r, im)
x = c.sum() #pass those 2 arrays, get to know sum of them both
print(f"The summed complex no. : {x[0]} + ({x[1]})j")

```

9. Develop a program that uses class Student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details.

```

class Student:
    def __init__(self, name, marks1, marks2, marks3):
        self.name = name
        self.marks1 = marks1
        self.marks2 = marks2
        self.marks3 = marks3

    def calculate_total(self):
        return self.marks1 + self.marks2 + self.marks3

    def calculate_percentage(self):
        total = self.calculate_total()
        return (total / 300) * 100

name = input("Enter student name: ")
marks = list(map(float, input("Enter marks in subject 1, 2 and 3: ").split(' ')))
student = Student(name, marks[0], marks[1], marks[2])
total_marks = student.calculate_total()
percentage = student.calculate_percentage()
print(f"Total marks: {total_marks}\nPercentage: {percentage: 0.3f}")

```

10. Demonstrates the concept of creating a BankAccount class in Python and using its methods to perform various account operations like deposit, withdraw and display balance

```
class BankAccount:
    def __init__(self, balance=0):
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        self.display_balance()

    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
        else:
            print("Insufficient balance")
        self.display_balance()

    def display_balance(self):
        print("Current Balance: Rs.", self.balance)

account = BankAccount()
account.deposit(1000)
account.withdraw(500)
account.display_balance()
```