

13-kaushal-khatiwada

November 21, 2024

0.1 Programming Exercises

NAME: KAUSHAL KHATIWADA ROLL NUMBER: 13

1 Q no 1

Write a program display “Hello World!”

```
[2]: print("Hello World!")
```

Hello World!

2 Q no 2

Write a program to find sum of two numbers. Use input() function to take input from the user.

```
[4]: a = int(input("Enter First number: "))
b = int(input("Enter Second number: "))
sum = a + b
print(f"Sum of two numbers {a} and {b} is {sum}")
```

Sum of two numbers 5 and 6 is 11

3 Q no 3

Write a program to calculate discount on the basis of following assumption: a) If purchased amount is greater than or equal to 1000, discount is 5%

```
[2]: try:
    Amount = float(input("Enter an Amount: "))
    Discount = 0
    if Amount < 0:
        print("Value of Amount must be positive")

    elif Amount >=1000:
        Discount = 0.05 * Amount
        print(f"Congratulation!! your discounted Amount is {Discount} for Rs_
↪{Amount}")
```

```

    else:
        print("Sorry, Try Again!!")
except ValueError:
    print("Value Error")

```

Congratulation!! your discounted Amount is 50.0 for Rs 1000.0

4 Q no 4

Write a program to calculate discount on the basis of following assumption: a) If purchased amount is greater than or equal to 1000, discount is 5% b) If purchased amount is less than 1000, discount is 3%

```

[10]: try:
    Amount = float(input("Enter an Amount: "))
    Discount = 0
    if Amount < 0:
        print("Value of Amount must be positive")

    elif Amount >= 1000:
        dis_perc = 5
        Discount = (dis_perc/100) * Amount

    else:
        dis_perc = 3
        Discount = (dis_perc/100) * Amount
    print(f"Congratulation!! You get discount of Rs {Discount} in your Rs_
↪{Amount} item at {dis_perc}%")

except ValueError:
    print("Value Error")

```

Congratulation!! You get discount of Rs 50.0 in your Rs 1000.0 item at 5%

5 Qno 5

Write a program to calculate discount on the basis of following assumption: a) If purchased amount is greater than or equal to 5000, discount is 10% b) If purchased amount is greater than or equal to 4000 and less than 5000, discount is 7% c) If purchased amount is greater than or equal to 3000 and less than 4000, discount is 5% d) If purchased amount is greater than or equal to 2000 and less than 3000, discount is 3% e) If purchased amount is less than 2000, discount is 2%

```

[18]: Amount = float(input("Enter an Amount: "))
Discount = 0
if Amount >= 5000:
    dis_perc = 10

```

```

    Discount = (dis_perc/100) * Amount
elif Amount >=4000:
    dis_perc = 7
    Discount = (dis_perc/100) * Amount
elif Amount >=3000:
    dis_perc = 5
    Discount = (dis_perc/100) * Amount
elif Amount >=2000:
    dis_perc = 3
    Discount = (dis_perc/100) * Amount
else:
    dis_perc = 2
    Discount = (dis_perc/100) * Amount
print(f"Congratulation!! You get discount of Rs {Discount} in your Rs {Amount}
↳item at {dis_perc}%")

```

Congratulation!! You get discount of Rs 20.0 in your Rs 1000.0 item at 2%

6 Q no 6

Write a program to calculate the simple interest on the basis of following assumption: a) If balance is greater than or equal to 100000, interest is 7 % b) If balance is greater than or equal to 50000 and less than 100000 interest is 5 % c) If balance is less than 50000, interest is 3%

```

[12]: Balance = float(input("How much is your Balance Amount: "))
Interest = 0
if Balance >= 100000:
    Rate =7
    Interest = (Rate/100) * Balance
elif 100000< Balance <= 50000:
    Rate = 5
    Interest = (Rate/100) * Balance
else:
    Rate = 3
    Interest = (Rate/100) * Balance

print(f"Your Interest of Balance Rs {Balance} will be {Interest} at {Rate}%")

```

Your Interest of Balance Rs 1000.0 will be 30.0 at 3%

7 Q no 7

Write a program to test whether a number entered is even or odd

```

[ ]: a = int(input("Enter any number: "))
if a%2==0:
    print("Your selected number is EVEN")

```

```
else:
    print("Your selected number is ODD")
```

Your selected number is ODD

8 Q no 8

Admission to a professional course is subject to the following conditions: a) Marks in mathematics ≥ 60 b) Marks in physics ≥ 50 c) Marks in chemistry ≥ 40 d) Total in all three subjects ≥ 200 OR Total in mathematics and physics ≥ 150 Given the marks in three subjects, write a program to process the applications to list eligible candidates.

```
[48]: math = int(input("Enter your Mathematics Marks:"))
physics = int(input("Enter your Physics Marks:"))
chemistry = int(input("Enter your Chemistry Marks:"))
if math >= 60 and physics >= 50 and chemistry >= 40:
    if math + physics >= 150:
        if math + physics + chemistry >= 200:
            print("Dear Student, You are eligible!!")
        else:
            print("Dear Student, You are not eligible as total of three subject_
↳ less than 200")
    else:
        print("Dear Student, You are not eligible as total of two subject (Math_
↳ and Physics) is less than 150")
else:
    print("Dear Student, You are not eligible as marks for any subject is less_
↳ than pass marks")
print(f"Mathematics:{math}    Physics:{physics}    Chemistry:{chemistry}")
print(f"Total of Math and Physics : {math+physics}")
print(f"Total : {math+physics+chemistry}")
```

```
Dear Student, You are eligible!!
Mathematics:80    Physics:80    Chemistry:80
Total of Math and Physics : 160
Total : 240
```

9 Q no 9

The rates of tax on gross salary are as shown below: Income Tax Less than 10,000 Nil Rs. 10,000 to 19,999 10% Rs. 20,000 to 39,999 15% Rs. 40,000 to above 20% Write a program to compute the net salary after deducting the tax for the given information and print the same.

```
[57]: Gross_Salary = float(input("Enter your Gross Salary: "))
if Gross_Salary >= 40000:
    Rate = 20
    Net_Salary = Gross_Salary - (Rate/100) * Gross_Salary
```

```

elif Gross_Salary >= 20000:
    Rate = 15
    Net_Salary = Gross_Salary - (Rate/100) * Gross_Salary
elif Gross_Salary >= 10000:
    Rate = 10
    Net_Salary = Gross_Salary - (Rate/100) * Gross_Salary
else:
    Rate = 0
    Net_Salary = Gross_Salary - (Rate/100) * Gross_Salary
print(f"Gross Salary: Rs {Gross_Salary}")
print(f"Tax Amount: Rs {Rate/100 * Gross_Salary} deducted at {Rate}%")
print(f"Net Salary: Rs {Net_Salary}")

```

Gross Salary: Rs 15000.0
Tax Amount: Rs 1500.0 deducted at 10%
Net Salary: Rs 13500.0

10 Q no 10

Write a program to check whether a year entered is leap or not

The main logic behind leap year detection are Year that can be exactly divided by 4 (such as 2020, 2024, 2028, etc) except if it can be exactly divided by 100, then it isn't (such as 2100, 2200, etc) except if it can be exactly divided by 400, then it is (such as 2000, 2400)

Why? Because the Earth rotates about 365.2422 times a year but a normal year is 365 days, so something has to be done to "catch up" the extra 0.2422 days a year.

Every 4th year we add an extra day (the 29th of February), which makes 365.25 days a year. This is fairly close, but is wrong by about 1 day every 100 years. So every 100 years we don't have a leap year, and that gets us 365.24 days per year (1 day less in 100 year = -0.01 days per year). Closer, but still not accurate enough! So another rule says that every 400 years is a leap year again. This gets us 365.2425 days per year (1 day regained every 400 years = 0.0025 days per year), which is close enough to 365.2422 not to matter much.

```

[4]: year = int(input("Enter a Year on AD: "))
if (year % 4 == 0):
    if (year % 100 == 0):
        if (year % 400 == 0):
            print(f"{year} is a leap year.")
        else:
            print(f"{year} is not a leap year.")    # Not Divisible by 400, not leap
    else:
        print(f"{year} is a leap year.")    # Not Divisible by 100, leap
else:
    print(f"{year} is not a leap year.")    # Not Divisible by 4, not leap

```

2025 is not a leap year.

11 Q no 11

Write a program using match-case statement to develop a simple calculator.

```
[7]: num1 = float(input("Enter the first number: "))
operator = input("Enter an operator (+, -, *, /): ")
num2 = float(input("Enter the second number: "))

match operator:
    case '+':
        result = num1 + num2
        print(f"{num1} + {num2} = {result}.")
    case '-':
        result = num1 - num2
        print(f"{num1} - {num2} = {result}.")
    case '*':
        result = num1 * num2
        print(f"{num1} * {num2} = {result}.")
    case '/':
        if num2 != 0:
            result = num1 / num2
            print(f"{num1} / {num2} = {result}.")
        else:
            print("Error: Division by zero is not allowed.")
    case _:
        print("Error: Invalid operator.")
```

5.0 + 7.0 = 12.0.

12 Q no 12

Write a program to display “MDS” 10 times.

```
[63]: for x in range(0,10):
        print (f"{x+1} MDS")
```

```
1 MDS
2 MDS
3 MDS
4 MDS
5 MDS
6 MDS
7 MDS
8 MDS
9 MDS
10 MDS
```

13 Q no 13

Write a program to find sum and average of 10 numbers stored in a list.

```
[ ]: nums = [1,2,3,4,5,6,7,8,9,10]

total = sum(nums)

average = total / len(nums)

print(f"Sum of the numbers: {total}")
print(f"Average of the numbers: {average}")
```

Sum of the numbers: 55

Average of the numbers: 5.5

14 Q no 14

Write a program to check whether a number entered is prime or not. Prime numbers are divisible only by 1 and itself

```
[13]: def is_prime(number):
        if number <= 1:
            return False
        for i in range(2, number):
            if number % i == 0:
                return False
        return True
```

```
[ ]: num = int(input("Enter a number: "))
if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is not a prime number.")
```

5 is a prime number.

15 Q no 15

Write a program to display prime numbers up to 100.

```
[15]: def is_prime(number):
        if number <= 1:
            return False
        for i in range(2, int(number**0.5) + 1):
            if number % i == 0:
                return False
        return True
```

```
[ ]: print("Prime numbers up to 100:")
    for num in range(2, 101):
        if is_prime(num):
            print(num, end=" ")
```

Prime numbers up to 100:

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

16 Q no 16

Write a program to find sum of digits of a number.

```
[ ]: def sum_of_digits(number):
    total = 0
    while number > 0:
        total += number % 10
        number //= 10
    return total
```

```
[ ]: num = int(input("Enter a number: "))
    result = sum_of_digits(abs(num))
    print(f"The sum of the digits of {num} is {result}.")
```

The sum of the digits of 234 is 9.

17 Q no 17

Write a program to check whether a number is palindrome or not.

```
[ ]: def is_palindrome(number):
    original_number = str(number)
    reversed_number = original_number[::-1]
    return original_number == reversed_number
```

```
[ ]: num = int(input("Enter a number: "))
    if is_palindrome(num):
        print(f"{num} is a palindrome.")
    else:
        print(f"{num} is not a palindrome.")
```

1123211 is a palindrome.

18 Q no 18

Write a program to check if a number is Armstrong Number or not. Number that is equal to the sum of its own digits, each raised to the power of the number of digits.


```
[ ]: def is_armstrong_number(num):

    digits = str(num)
    num_digits = len(digits)

    armstrong_sum = sum(int(digit) ** num_digits for digit in digits)

    return armstrong_sum == num

[ ]: number = int(input("Enter a number to check if it's an Armstrong number: "))

if is_armstrong_number(number):
    print(f"{number} is an Armstrong number.")
else:
    print(f"{number} is not an Armstrong number.")
```

132 is not an Armstrong number.

19 Q no 19

Write a program to count number of vowels in a string.

```
[ ]: def count_vowels(string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in string:
        if char in vowels:
            count += 1
    return count

[ ]: text = input("Enter a string: ")
vowel_count = count_vowels(text)
print(f"The number of vowels in the string '{text}' is: {vowel_count}")
```

The number of vowels in the string 'kaushal' is: 3

20 Q no 20

Write a program to find smallest and largest number among 10 numbers stored in a list.

```
[43]: def find_smallest_and_largest(numbers):
    smallest = min(numbers)
    largest = max(numbers)
    return smallest, largest

[ ]: print("Enter 10 numbers:")
num_list = [int(input()) for _ in range(10)]
```

```
smallest, largest = find_smallest_and_largest(num_list)
print(f"The smallest number is: {smallest}")
print(f"The largest number is: {largest}")
```

Enter 10 numbers:

The smallest number is: 1

The largest number is: 10

21 Q no 21

Write a program to count even numbers and odd numbers stored in a list.

```
[46]: def count_even_odd(numbers):
        even_count = 0
        odd_count = 0
        for num in numbers:
            if num % 2 == 0:
                even_count += 1
            else:
                odd_count += 1
        return even_count, odd_count
```

```
[ ]: print("Enter numbers separated by spaces:")
num_list = list(map(int, input().split()))

even_count, odd_count = count_even_odd(num_list)
print(f"Number of even numbers: {even_count}")
print(f"Number of odd numbers: {odd_count}")
```

Enter numbers separated by spaces:

Number of even numbers: 5

Number of odd numbers: 5

22 Q no 22

Write a program to find sum of two matrices.

```
[5]: matrix1 = [[1, 2],
                [3, 4]]

matrix2 = [[5, 6],
           [7, 8]]

result = [[0, 0],
          [0, 0]]
```

```

for i in range(len(matrix1)):
    for j in range(len(matrix1[0])):
        result[i][j] = matrix1[i][j] + matrix2[i][j]

print("Sum of 2X2 matrices:")
for row in result:
    print(row)

```

Sum of 2X2 matrices:
[6, 8]
[10, 12]

23 Q no 23

Write a program to find product of two matrices.

```

[7]: matrix1 = [[1, 2, 3],
                [4, 5, 6]]

matrix2 = [[1, 4],
           [2, 5],
           [3, 6]]

result = [[0, 0],
          [0, 0]]

for i in range(len(matrix1)):
    for j in range(len(matrix2[0])):
        for k in range(len(matrix2)):
            result[i][j] += matrix1[i][k] * matrix2[k][j]

print("Product of matrices:")
for row in result:
    print(row)

```

Product of matrices:
[14, 32]
[32, 77]

24 Q no 24

Write a program using list comprehension to find sum of only even numbers.

```

[8]: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_sum = sum([num for num in numbers if num % 2 == 0])
print("Sum of even numbers:", even_sum)

```

Sum of even numbers: 30

25 Q no 25

Write a program using function with return type to find sum of two numbers.

```
[9]: def sum_of_two_numbers(num1, num2):  
    return num1 + num2  
  
number1 = float(input("Enter the first number: "))  
number2 = float(input("Enter the second number: "))  
  
result = sum_of_two_numbers(number1, number2)  
  
print(f"The sum of {number1} and {number2} is: {result}")
```

The sum of 5.0 and 6.0 is: 11.0

26 Q no 26

Write a program using recursive function to find factorial of a number.

```
[ ]: def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n - 1)  
  
[ ]: number = int(input("Enter a number to find its factorial: "))  
result = factorial(number)  
print(f"The factorial of {number} is: {result}")
```

The factorial of 6 is: 720

27 Q no 27

Write a program using recursive function to find nth Fibonacci number.

```
[ ]: def fibonacci(n):  
    if n <= 1:  
        return n  
    else:  
        return fibonacci(n - 1) + fibonacci(n - 2)  
  
n = int(input("Enter the position of the Fibonacci number to find: "))  
  
result = fibonacci(n)
```

```
print(f"The {n}th Fibonacci number is: {result}")
```

The 6th Fibonacci number is: 8

28 Q no 28

Create a class Rectangle containing instance variables length and breadth. The class also contains two instance methods area() and perimeter() to find area and perimeter of rectangles respectively. Use this class to find area and perimeter of two different rectangles.

```
[1]: class Rectangle:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def area(self):
        a = self.length * self.breadth
        print(f"Area of a rectangle with length {self.length} and breadth {self.breadth} is {a}")

    def perimeter(self):
        p = 2 * (self.length + self.breadth)
        print(f"Perimeter of a rectangle with length {self.length} and breadth {self.breadth} is {p}")
```

```
[3]: rect1 = Rectangle(4,2)
rect1.area()
rect1.perimeter()
```

Area of a rectangle with length 4 and breadth 2 is 8
Perimeter of a rectangle with length 4 and breadth 2 is 12

```
[4]: rect2 = Rectangle(8,4)
rect2.area()
rect2.perimeter()
```

Area of a rectangle with length 8 and breadth 4 is 32
Perimeter of a rectangle with length 8 and breadth 4 is 24

29 Q no 29

Create a class Circle containing an instance variable radius. The class also contains two instance methods area() and circumference() to find area and circumference of circles respectively. Use this class to find area and circumference of two different circles. Use PI as a class variable.

```
[7]: class Circle:
    PI = 3.14          # class varibale
    def __init__(self,radius):
        self.radius = radius

    def area(self):
        a = Circle.PI * self.radius ** 2
        print(f"Area of circle with {self.radius} is {a}")

    def cicumference(self):
        c = 2 * Circle.PI * self.radius
        print(f"Circumference of circle with {self.radius} is {c}")
```

```
[10]: circ1 = Circle(3)
      circ1.area()
      circ1.cicumference()
```

Area of circle with 3 is 28.26
Circumference of circle with 3 is 18.84

```
[9]: circ2 = Circle(4)
      circ2.area()
      circ2.cicumference()
```

Area of circle with 4 is 50.24
Circumference of circle with 4 is 25.12

30 Q no 30

Create a class Box with instance variables length, width and height. The class also contains instance methods volume() and surface_area() to find volume and surface area of boxes respectively. Use this class to find volume and surface area of two different boxes.

```
[13]: class Box:
    def __init__(self,length,width,height):
        self.length = length
        self.width = width
        self.height = height

    def volume(self):
        v = self.length * self.width * self.height
        print(f"Volume of Box is {v}")

    def surface_area(self):
        sa = 2 * self.length * self.height + 2 * self.length * self.width + 2 *
        ↪self.width * self.height
        print(f"Surface Area of Box is {sa}")
```

```
[14]: box1 = Box(10,5,4)
      box1.volume()
      box1.surface_area()
```

Volume of Box is 200
Surface Area of Box is 220

```
[16]: box2 = Box(5,8,2)
      box2.volume()
      box2.surface_area()
```

Volume of Box is 80
Surface Area of Box is 132

31 Q no 31

Create a class Time with three instance variables hours, minutes, and seconds. Add instance methods display() to display the time in hh:mm:ss format and add() to add two time objects. Use this class to add and display two different time objects.

```
[13]: class Time:
      def __init__(self, hours, minutes, seconds):
          self.hours = hours
          self.minutes = minutes
          self.seconds = seconds

      def add(self, other):
          total_seconds = (self.hours * 3600) + (self.minutes * 60) + self.
↪seconds + \
                                (other.hours * 3600) + (other.minutes * 60) + other.
↪seconds
          self.hours = total_seconds // 3600
          self.minutes = (total_seconds % 3600) // 60
          self.seconds = total_seconds % 60

      def display(self):
          print(f"Your Time is {self.hours}:{self.minutes}:{self.seconds}")
```

```
[14]: time1 = Time(12,35,59)
      time2 = Time(12,12,1)

      time1.add(time2)
      time1.display()
```

Your Time is 24:48:0

32 Q no 32

Create a class Distance containing instance variables feet and inches. The class also contains instance methods add() and compare() to add and compare two distance objects respectively. Use this class to create two different distance objects and add and compare these two distance objects.

```
[27]: class Distance:
    def __init__(self, feet, inches):
        self.feet = feet
        self.inches = inches

    def add(self, other):
        total_inches = self.inches + other.inches
        extra_feet = total_inches // 12
        total_feet = self.feet + other.feet + extra_feet
        remaining_inches = total_inches % 12
        return Distance(total_feet, remaining_inches)

    def compare(self, other):
        total_inches_self = self.feet * 12 + self.inches
        total_inches_other = other.feet * 12 + other.inches
        if total_inches_self > total_inches_other:
            return f"Distance 1 ({self.feet} feet {self.inches} inches) is_
↳greater."
        elif total_inches_self < total_inches_other:
            return f"Distance 2 ({other.feet} feet {other.inches} inches) is_
↳greater."
        else:
            return "Both distances are equal."

    def __str__(self):
        return f"{self.feet} feet {self.inches} inches"
```

```
[29]: # Create two distance objects
d1 = Distance(5, 8)
d2 = Distance(3, 10)

# Add the two distances
d3 = d1.add(d2)
print("Sum of distances:", d3)
```

Sum of distances: 9 feet 6 inches

```
[30]: # Compare the two distances
result = d1.compare(d2)
print(result)
```

Distance 1 (5 feet 8 inches) is greater.

33 Q no 33

Create a class Student with instance variables name, roll_number, and marks in five subjects. Add three instance methods in this class to calculate total(), percentage(), and division() of the marks obtained by the students. Use this class to find total marks obtained, percentage, and division of five students.

```
[1]: class Student:
    def __init__(self, name, roll_number, marks):
        self.name = name
        self.roll_number = roll_number
        self.marks = marks          # List of marks in 5 subjects

    def total(self):
        return sum(self.marks)

    def percentage(self):
        return self.total() / len(self.marks)

    def division(self):
        percentage = self.percentage()
        if percentage >= 80:
            return "Distinction"
        elif percentage >= 60:
            return "First Division"
        elif percentage >= 50:
            return "Second Division"
        elif percentage >= 40:
            return "Third Division"
        else:
            return "Fail"

    def __str__(self):
        return f"{self.roll_number} {self.name}, Total Marks: {self.total()},  
↳Percentage: {self.percentage():.2f}%, Division: {self.division()}"
```

```
[2]: # Create five student objects
students = [
    Student("Abin", 1, [85, 78, 92, 88, 76]),
    Student("Anish", 2, [72, 64, 58, 60, 70]),
    Student("Ankit", 3, [40, 45, 38, 42, 50]),
    Student("Ashmita", 4, [90, 95, 88, 92, 85]),
    Student("Atul", 5, [55, 58, 60, 53, 57]),
]
```

```
[3]: # Find total marks, percentage, and division of each student
for student in students:
    print(student)
```

1 Abin, Total Marks: 419, Percentage: 83.80%, Division: Distinction
 2 Anish, Total Marks: 324, Percentage: 64.80%, Division: First Division
 3 Ankit, Total Marks: 215, Percentage: 43.00%, Division: Third Division
 4 Ashmita, Total Marks: 450, Percentage: 90.00%, Division: Distinction
 5 Atul, Total Marks: 283, Percentage: 56.60%, Division: Second Division

34 Q no 34 (Inheritance)

Create a parent class Bonus with instance variables sales_id and sales_amount. Add get_bonus method that calculates a salesperson's bonus using the formula $\text{bonus} = \text{sales} * 0.05$. Create a child class named PremiumBonus from Bonus. The child class's get_premium_bonus() method should calculate the bonus using the formula $\text{bonus} = \text{sales} * 0.05 + (\text{sales} - 2500) * 0.01$. Now, create an object of PremiumBonus class and use this object to find both bonus and premium bonus.

```
[20]: # Parent Class: Bonus
class Bonus:
    def __init__(self, sales_id, sales_amount):
        self.sales_id = sales_id
        self.sales_amount = sales_amount

    # Method to calculate the bonus
    def get_bonus(self):
        return self.sales_amount * 0.05

# Child Class: PremiumBonus, inheriting from Bonus
class PremiumBonus(Bonus):
    def __init__(self, sales_id, sales_amount):
        super().__init__(sales_id, sales_amount)

    # Method to calculate the premium bonus
    def get_premium_bonus(self):
        bonus = self.sales_amount * 0.05
        if self.sales_amount > 2500:
            bonus += (self.sales_amount - 2500) * 0.01
        return bonus

# Create an object of PremiumBonus class
premium_salesperson = PremiumBonus(sales_id=101, sales_amount=3000)

# Calculate and print both bonus and premium bonus
bonus = premium_salesperson.get_bonus()
premium_bonus = premium_salesperson.get_premium_bonus()

print(f"Sales ID: {premium_salesperson.sales_id}")
print(f"Sales Amount: {premium_salesperson.sales_amount}")
print(f"Bonus: {bonus}")
print(f"Premium Bonus: {premium_bonus}")
```

Sales ID: 101
Sales Amount: 3000
Bonus: 150.0
Premium Bonus: 155.0

35 Q no 35

Overload + operator in Q.N.25 to add two distance objects.

```
[15]: class Distance:
    def __init__(self, feet, inches):
        self.feet = feet
        self.inches = inches

    def __add__(self, other):
        total_inches = self.inches + other.inches
        total_feet = self.feet + other.feet + \
            total_inches // 12 #
        return Distance(total_feet, total_inches % 12)

    def __str__(self):
        return f"{self.feet} feet {self.inches} inches"
```

```
[17]: distance1 = Distance(5, 8)
distance2 = Distance(5, 9)

result = distance1 + distance2

print("Sum of distances:", result)
```

Sum of distances: 11 feet 5 inches

36 Q no 36

Write a program that reads a text file and displays the following: a. Number of characters b. Number of vowels c. Number of consonants d. Number of words e. Number of lines

```
[23]: def count_text_file_details(filename):
    try:
        with open(filename, 'r') as file:
            text = file.read()

            # count total characters
            num_characters = len(text)

            # count vowels
            num_vowels = sum(1 for char in text.lower() if char in 'aeiou')
```

```

    # count consonants
    num_consonants = sum(1 for char in text if char.isalpha()
                        and char.lower() not in 'aeiou')

    # count words
    num_words = len(text.split())

    # count lines
    num_lines = text.count('\n') + 1

    # Display the results
    print(f"Number of characters: {num_characters}")
    print(f"Number of vowels: {num_vowels}")
    print(f"Number of consonants: {num_consonants}")
    print(f"Number of words: {num_words}")
    print(f"Number of lines: {num_lines}")

except FileNotFoundError:
    print(f"The file {filename} was not found!")

```

```

[25]: filename = input("Enter the name of the text file: ")
      count_text_file_details(filename)

```

```

Number of characters: 54
Number of vowels: 15
Number of consonants: 25
Number of words: 10
Number of lines: 3

```

37 Q no 37

Write a program that reads a text file and writes its output in another text file. The output should contain a. Number of letters b. Number of digits, and c. Number of other characters

```

[26]: def count_file_characters(input_filename, output_filename):
      try:
          with open(input_filename, 'r') as input_file:
              text = input_file.read() # Read the entire content of the file

              # count letters
              num_letters = sum(1 for char in text if char.isalpha())

              # count digits
              num_digits = sum(1 for char in text if char.isdigit())

              # count other characters (excluding letters and digits)

```

```

        num_other_chars = len(text) - num_letters - num_digits

        # Prepare the output text
        output_text = f"Number of letters: {num_letters}\n"
        output_text += f"Number of digits: {num_digits}\n"
        output_text += f"Number of other characters: {num_other_chars}\n"

        # Write the result to the output file
        with open(output_filename, 'w') as output_file:
            output_file.write(output_text)

        print(f"Results have been written to {output_filename}")

    except FileNotFoundError:
        print(f"The file {input_filename} was not found!")

```

```

[27]: input_filename = input("Enter the name of the input text file: ")
      output_filename = input("Enter the name of the output text file: ")
      count_file_characters(input_filename, output_filename)

```

Results have been written to count.txt

Content of Output File Number of letters: 40 Number of digits: 2 Number of other characters: 12

38 Q no 38

Write a program that reads the file containing texts and counts the number of whitespaces.

```

[31]: def count_whitespace_characters(filename):
      try:
          with open(filename, 'r') as file:
              text = file.read()
              # Count all whitespace characters in the line
              num_whitespaces = sum(1 for char in text if char.isspace())

              # Display the result
              print(f"Number of whitespace characters: {num_whitespaces}")

      except FileNotFoundError:
          print(f"The file {filename} was not found!")

```

```

[32]: # Example usage:
      filename = input("Enter the name of the text file: ")
      count_whitespace_characters(filename)

```

Number of whitespace characters: 13

39 Q no 39

Write a program to find sum and average of numbers stored in a file. Create a separate file to write output.

```
[26]: def calculate_sum_and_average(input_filename, output_filename):
    try:
        with open(input_filename, 'r') as input_file:
            # Read all lines and convert them to a list of numbers
            numbers = [float(line.strip()) for line in input_file if line.
↪strip().isdigit() or is_float(line.strip())]

            if numbers: # Check if the list is not empty
                total_sum = sum(numbers)
                average = total_sum / len(numbers)

                # Prepare output text
                output_text = f"Sum of numbers: {total_sum}\n"
                output_text += f"Average of numbers: {average}\n"

                # Write to the output file
                with open(output_filename, 'w') as output_file:
                    output_file.write(output_text)

                print(f"Sum and Average have been written to {output_filename}")
            else:
                print("No valid numbers found in the input file.")

    except FileNotFoundError:
        print(f"The file {input_filename} was not found!")

def is_float(value):
    try:
        float(value)
        return True
    except ValueError:
        return False

# Example usage:
input_filename = input("Enter the name of the input text file: ")
output_filename = input("Enter the name of the output text file: ")
calculate_sum_and_average(input_filename, output_filename)
```

Sum and Average have been written to 39_count.txt

40 Q no 40 (Numpy)

Write a program to find sum and average of 10 numbers stored in a numpy array.

```
[36]: import numpy as np
```

```
[37]: def sum_and_average():  
  
    numbers = np.random.random(10)  
    total = np.sum(numbers)  
    average = np.mean(numbers)  
  
    print("Sum:", total)  
    print("Average:", average)
```

```
[38]: sum_and_average()
```

Sum: 4.0888253462843505

Average: 0.40888253462843505

41 Q no 41 (Numpy)

Write a program to find sum of each row of a matrix stored in a numpy array.

```
[49]: def sum_of_rows():  
  
    matrix = np.array([[1, 2, 3],  
                        [4, 5, 6],  
                        [7, 8, 9]])  
  
    row_sums = np.sum(matrix, axis=1)  
    print("Sum of each row:")  
    for i, row_sum in enumerate(row_sums):  
        print(f"Row {i + 1}: {row_sum}")  
  
sum_of_rows()
```

Sum of each row:

Row 1: 6

Row 2: 15

Row 3: 24