#python program to calculate the sum, difference, product and quotient of two numbers

"""

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

sum = num1 + num2

difference = num1 - num2

product = num1 \* num2

quotient= num1 / num2  # This will raise an error if num2 is 0

print("Sum:", sum)

print("Difference:", difference)

print("Product:", product)

print("Quotient:", quotient)"""

#python program to calculate the remainder and power of two numbers

"""num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

remainder = num1 % num2

power = num1 \*\* num2

print("Remainder:", remainder)

print("Power:", power)"""

#Given two variables 'a' and 'b',write a program to check if a greater than, less than,equal to or not equal to b

"""a = float(input("Enter the value for a: "))

b = float(input("Enter the value for b: "))

print("a is greater than b" \*(a > b))

print("a is less than b" \*(a < b))

print("a is equal to b" \*(a == b))

print("a is not equal to b" \*(a != b))"""

#Write a python program to compare two strings and print whether they are the same or different

"""string1 = input("Enter the first string: ")

string2 = input("Enter the second string: ")

print("The strings are the same" \* (string1 == string2) or "The strings are different")"""

#Write a python program to check if a number is between 10 and 20 using logical operators

"""number = float(input("Enter a number: "))

print(10 <= number <= 20)"""

#Perform bitwise AND, OR, XOR, and NOT operations on two integers

'''

a = 10

b = 7

print("a =", a, "b =", b)

print("Bitwise AND:", a & b)

print("Bitwise OR:", a | b)

print("Bitwise XOR:", a ^ b)

print("Bitwise NOT a:", ~a)

print("Bitwise NOT b:", ~b)'''

#Write a program that checks if a number is positive, negative, or zero.

"""number = float(input("Enter a number: "))

if number > 0:

    print("The number is positive.")

else:

    if number < 0:

        print("The number is negative.")

    else:

        print("The number is zero.")"""

# Write a program to check if a year is a leap year.

"""year = int(input("Enter a year: "))

if year % 400 == 0:

    print("it  is a leap year")

else:

    if year % 100 == 0:

        print("It is not a leap year")

    else:

        if year % 4 == 0:

            print("it  is a leap year")

        else:

            print("It is not a leap year")"""

# Write a program to find the largest among three numbers using nested if statements.

'''

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

num3 = float(input("Enter the third number: "))

if num1 >= num2:

    if num1 >= num3:

        largest = num1

    else:

        largest = num3

else:

    if num2 >= num3:

        largest = num2

    else:

        largest = num3

print(f"The largest number is {largest}.")'''

 # Write a program to determine the grade of a student based on their score using nested if statements

'''score = float(input("Enter the student's score: "))

if 0 <= score <= 100:

    if score >= 90:

        grade = 'A'

    elif score >= 80:

        grade = 'B'

    elif score >= 70:

        grade = 'C'

    elif score >= 60:

        grade = 'D'

    else:

        grade = 'F'

    # Print the result

    print(f"The grade for a score of {score} is: {grade}")

else:

    # Handle invalid score

    print("Invalid score. Please enter a score between 0 and 100.")'''

#Create a list with 5 elements and demonstrate adding, removing, and accessing elements.

'''

my\_list = [1, 2, 3, 4, 5]

print("Original list:", my\_list)

print("First element:", my\_list[0])

print("Last element:", my\_list[-1])

my\_list.append(6)

print("List after adding 6:", my\_list)

my\_list.remove(3)

print("List after removing 3:", my\_list)

removed\_element = my\_list.pop(2)

print("Removed element:", removed\_element)

print("List after popping the element at index 2:", my\_list)'''

#Write a program to reverse a list and print the reversed list.

'''my\_list = [1, 2, 3, 4, 5]

print("Original list:", my\_list)

my\_list.reverse()

print("Reversed list:", my\_list)'''

# Given a list, write a program to print the first three elements, the last three elements, and elements from index from index 2 to 4

'''my\_list = [10, 11, 12, 99, 101, 1100, 19, 80, 70, 60,50]

first\_three = my\_list[:3]

print("First three elements:", first\_three)

last\_three = my\_list[-3:]

print("Last three elements:", last\_three)

elements\_2\_to\_4 = my\_list[2:5]

print("Elements from index 2 to 4:", elements\_2\_to\_4)'''

#Write a program to create a sublist from a list using slicing

'''original\_list =  [10, 11, 12, 99, 101, 1100, 19, 80, 70, 60,50]

sublist = original\_list[2:5]

print("Original list:", original\_list)

print("Sublist:", sublist)'''

#Write a program to sort a list of integers in ascending order and find the index of a specific value.

'''numbers = [1, 2, 7, 10, 99, 69, 75, 89, 34, 32]

numbers.sort()

print("Sorted list:", numbers)

value\_to\_find = 69

if value\_to\_find in numbers:

    print(f"The index of {value\_to\_find} is:", numbers.index(value\_to\_find))

else:

    print(f"{value\_to\_find} is not in the list.")'''

#Write a program to append elements to a list and then remove an element from the list.

'''my\_list = [10, 20, 30]

print("Original list:", my\_list)

my\_list.append(40)

my\_list.append(50)

print("List after appending elements:", my\_list)

my\_list.remove(30)

print("List after removing 30:", my\_list)'''

#Create a tuple with 5 elements and demonstrate accessing elements.

'''my\_tuple = (10, 20, 30, 40, 50)

print("First element:", my\_tuple[0])

print("Second element:", my\_tuple[1])

print("Third element:", my\_tuple[2])

print("Fourth element:", my\_tuple[3])

print("Fifth element:", my\_tuple[4])

print("Last element:", my\_tuple[-1])

print("Second to last element:", my\_tuple[-2])'''

#Write a program to create a tuple of tuples and access individual elements.

'''tuple\_of\_tuples = ((1, 2, 3), (4, 5, 6), (7, 8, 9))

print("First tuple:", tuple\_of\_tuples[0])        # Access the first tuple

print("Second tuple:", tuple\_of\_tuples[1])       # Access the second tuple

print("Third tuple:", tuple\_of\_tuples[2])        # Access the third tuple

print("Element at (0, 0):", tuple\_of\_tuples[0][0])  # Access the first element of the first tuple

print("Element at (1, 2):", tuple\_of\_tuples[1][2])  # Access the third element of the second tuple

print("Element at (2, 1):", tuple\_of\_tuples[2][1])  # Access the second element of the third tuple'''

#Write a program to concatenate two tuples and find the length of the resulting tuple.

'''tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

concatenated\_tuple = tuple1 + tuple2

length\_of\_tuple = len(concatenated\_tuple)

print("Concatenated tuple:", concatenated\_tuple)

print("Length of the concatenated tuple:", length\_of\_tuple)'''

#Write a program to find the maximum and minimum values in a tuple.

'''my\_tuple = (10, 20, 4, 45, 99, 1)

print("Maximum value:", max(my\_tuple))

print("Minimum value:", min(my\_tuple))'''

#Create a dictionary with 3 key-value pairs and demonstrate adding, removing, and accessing elements.

'''my\_dict = {'name': 'Shivam', 'age': 25, 'city': 'New Delhi'}

print("Name:", my\_dict['name'])

print("Age:", my\_dict['age'])

print("City:", my\_dict['city'])

my\_dict['occupation'] = 'DevOps Enginner'

print("After adding occupation:", my\_dict)

del my\_dict['age']

print("After removing age:", my\_dict)'''

#Write a program to check if a key exists in a dictionary and print the corresponding value.

'''my\_dict = {'name': 'Shivam', 'age': 25, 'city': 'New Delhi'}

# Define the key to check

key\_to\_check = 'age'

# Check if the key exists in the dictionary

if key\_to\_check in my\_dict:

    print(f"The value for the key '{key\_to\_check}' is: {my\_dict[key\_to\_check]}")

else:

    print(f"Key '{key\_to\_check}' not found in the dictionary.")'''

#Write a program to update the value of an existing key and print all keys and values.

'''my\_dict = {'name': 'Sangram', 'age': 24, 'city': 'Uttar Pradesh'}

my\_dict['age'] = 30

for key, value in my\_dict.items():

    print(f"{key}: {value}")'''

#  Write a program to iterate through a dictionary and print all keys and values.

'''my\_dict = {'name': 'Aditya', 'age': 23, 'city': 'Himacha Pradesh'}

for key in my\_dict:

    print(f"{key}: {my\_dict[key]}")'''

#Create a set with 5 elements and demonstrate adding and removing elements.

'''my\_set = {1, 2, 3, 4, 5}

print("Original set:", my\_set)

my\_set.add(6)

print("Set after adding 6:", my\_set)

my\_set.remove(3)

print("Set after removing 3:", my\_set)

my\_set.discard(10)

print("Set after attempting to remove 10:", my\_set)'''

#Write a program to check if an element exists in a set.

'''

my\_set = {1, 2, 3, 4, 5}

element\_to\_check = 3

if element\_to\_check in my\_set:

    print(f"Element {element\_to\_check} exists in the set.")

else:

    print(f"Element {element\_to\_check} does not exist in the set.")'''

#Write a program to find the union, intersection, and difference between two sets.

'''

set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7, 8}

union\_set = set1 | set2

print("Union of set1 and set2:", union\_set)

intersection\_set = set1 & set2

print("Intersection of set1 and set2:", intersection\_set)

difference\_set = set1 - set2

print("Difference between set1 and set2:", difference\_set)

difference\_set2 = set2 - set1

print("Difference between set2 and set1:", difference\_set2)'''

#Write a program to find the symmetric difference between two sets

'''

set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7, 8}

symmetric\_difference = set1 ^ set2

print("Symmetric difference:", symmetric\_difference)'''

#Write a list comprehension to create a list of squares of numbers from 1 to 10

'''

squares = [x\*\*2 for x in range(1, 11)]

print(squares)'''

#Write a list comprehension to create a list of the first letters of each word in a given list of words.

'''

words = ['Messi', 'Ronaldo', 'Mbappe', 'Ronaldhino']

first\_letters = [word[0] for word in words]

print(first\_letters)'''

#Write a list comprehension to create a list of even numbers from 1 to 20

'''even\_numbers = [x for x in range(1, 21) if x % 2 == 0]

print(even\_numbers)'''

#Write a list comprehension to create a list of numbers from 1 to 20 that are divisible by 3.

'''divisible\_by\_3 = [x for x in range(1, 21) if x % 3 == 0]

print(divisible\_by\_3)'''

# Write a dictionary comprehension to create a dictionary where the keys are numbers from 1 to 5 and the values are their squares.

'''squares\_dict = {x: x\*\*2 for x in range(1, 6)}

print(squares\_dict)'''

# Write a dictionary comprehension to create a dictionary from two lists, one of keys and one of values.

'''keys = ['name', 'age', 'country']

values = ['Messi', 37, 'Argentina']

dictionary = dict(zip(keys, values))

print(dictionary)'''

#Write a dictionary comprehension to create a dictionary of numbers from 1 to 10 with values being 'even' or 'odd' based on the number's parity.

'''number\_parity = {x: 'even' if x % 2 == 0 else 'odd' for x in range(1, 11)}

print(number\_parity)'''

#Write a dictionary comprehension to filter out items from a dictionary where the value is less than 5

'''original\_dict = {'a': 3, 'b': 7, 'c': 1, 'd': 10, 'e': 4}

filtered\_dict = {key: value for key, value in original\_dict.items() if value >= 5}

print(filtered\_dict)'''