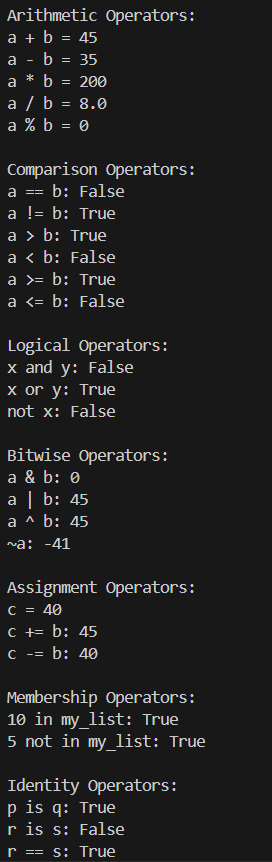
Output 1 :-



1. **Write a program to demonstrate different types of operators.**

a = 20

b = 4

print("Arithmetic Operators:")

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

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

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

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

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

print("\nComparison Operators:")

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

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

print("a > b:", a > b)

print("a < b:", a < b)

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

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

print("\nLogical Operators:")

x = True

y = False

print("x and y:", x and y)

print("x or y:", x or y)

print("not x:", not x)

print("\nBitwise Operators:")

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

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

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

print("~a:", ~a)

print("\nAssignment Operators:")

c = a

print("c =", c)

c += b

print("c += b:", c)

c -= b

print("c -= b:", c)

print("\nMembership Operators:")

my\_list = [1, 2, 3, 10]

print("10 in my\_list:", 10 in my\_list)

print("5 not in my\_list:", 5 not in my\_list)

print("\nIdentity Operators:")

p = 5

q = 5

r = [1, 2, 3]

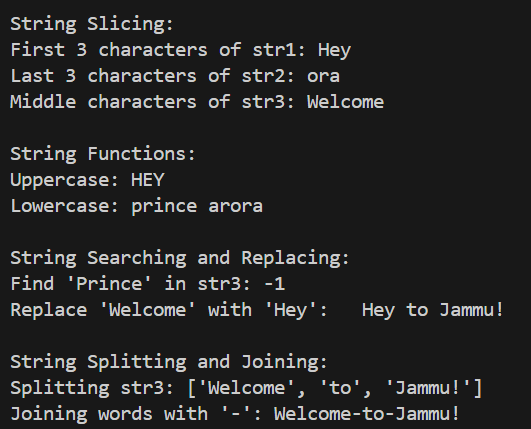
s = [1, 2, 3]

print("p is q:", p is q)

print("r is s:", r is s)

print("r == s:", r == s)

Output 2 :-



1. **Write a program to demonstrate string operations and functions.**

str1 = "Hey"

str2 = "Prince Arora"

str3 = "Welcome to Jammu!"

print("\nString Slicing:")

print("First 3 characters of str1:", str1[:3])

print("Last 3 characters of str2:", str2[-3:])

print("Middle characters of str3:", str3[2:10])

print("\nString Functions:")

print("Uppercase:", str1.upper())

print("Lowercase:", str2.lower())

print("\nString Searching and Replacing:")

print("Find 'Prince' in str3:", str3.find("Prince"))

print("Replace 'Welcome' with 'Hey':", str3.replace("Welcome", "Hey"))

print("\nString Splitting and Joining:")

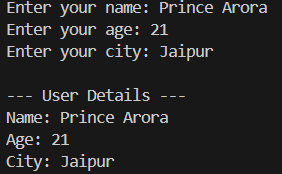
words = str3.split()

print("Splitting str3:", words)

joined\_str = "-".join(words)

print("Joining words with '-':", joined\_str)

Output 3 :-



1. **Write a program to take input from keyboard.**

name = input("Enter your name: ")

age = int(input("Enter your age: "))

city = input("Enter your city: ")

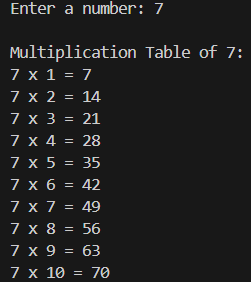
print("\n--- User Details ---")

print("Name:", name)

print("Age:", age)

print("City:", city)

Output 4 :-



1. **Write a program to implement For loop.**

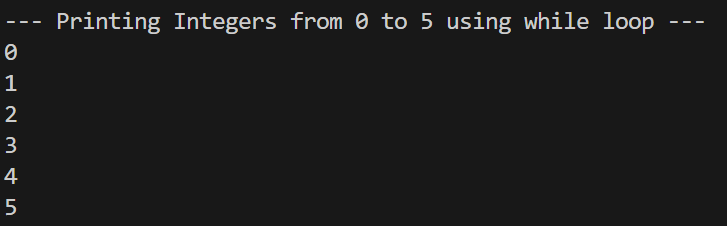
num = int(input("Enter a number: "))

print(f"\nMultiplication Table of {num}:")

for i in range(1, 11):

    print(f"{num} x {i} = {num \* i}")

Output 5 :-



1. **Write a program to print integer from 0 to 5 using while loop.**

print("\n--- Printing Integers from 0 to 5 using while loop ---")

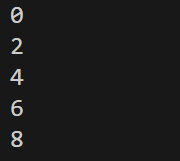
i = 0

while i <= 5:

    print(i)

    i += 1

Output 6 :-



1. **Write a program to print first five even number using break and continue statements.**

count = 0

num = 0

while True:

    if num % 2 != 0:

        num += 1

        continue

    print(num)

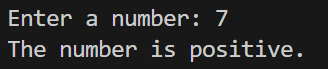
    count += 1

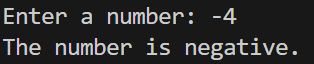
    if count == 5:

        break

    num += 1

Output 7 :-





1. **Write a program to implement if, elif and else statements.**

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

if num > 0:

    print("The number is positive.")

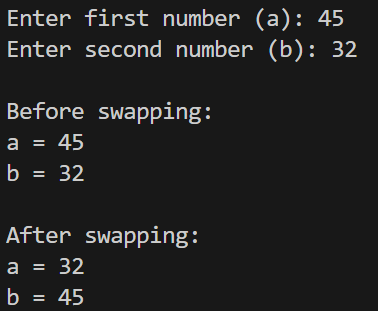
elif num < 0:

    print("The number is negative.")

else:

    print("The number is zero.")

Output 8 :-



1. **Write a program to swap the number of two variables.**

a = int(input("Enter first number (a): "))

b = int(input("Enter second number (b): "))

print("\nBefore swapping:")

print("a =", a)

print("b =", b)

temp = a

a = b

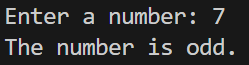
b = temp

print("\nAfter swapping:")

print("a =", a)

print("b =", b)

Output 9 :-





1. **Write a program to find whether a number is even or odd.**

num = int(input("Enter a number: "))

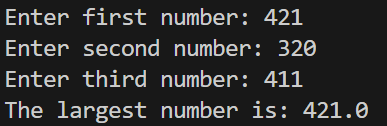
if num % 2 == 0:

    print("The number is even.")

else:

    print("The number is odd.")

Output 10 :-



1. **Write a program to check largest among the given three numbers.**

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

b = float(input("Enter second number: "))

c = float(input("Enter third number: "))

if a >= b and a >= c:

    print("The largest number is:", a)

elif b >= a and b >= c:

    print("The largest number is:", b)

else:

    print("The largest number is:", c)

Output 11 :-



1. **Write a program to display the Fibonacci sequence of n terms.**

n = int(input("Enter Number of terms: "))

a,b=0,1

if n<=0:

    for i in range(n):

        print(a,end=" ")

        nth=a-b

        a=b

        b=nth

elif n==1:

    print(" ",a," ")

else:

    for i in range(n):

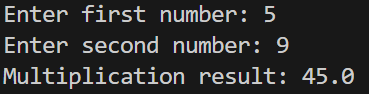
        print(a,end=" ")

        nth=a+b

        a=b

        b=nth

Output 12 :-



1. **Write a program for multiplication of two numbers using function.**

def multiply(x, y):

  return x \* y

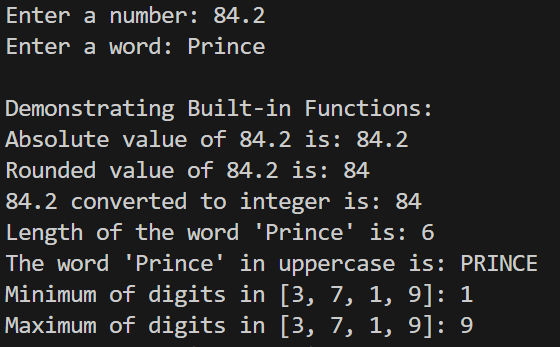
num1 = float(input("Enter first number: "))

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

result = multiply(num1, num2)

print("Multiplication result:", result)

Output 13 :-



1. **Write a program that demonstrates the build in functions.**

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

word = input("Enter a word: ")

print("\nDemonstrating Built-in Functions:")

print(f"Absolute value of {num} is:", abs(num))

print(f"Rounded value of {num} is:", round(num))

print(f"{num} converted to integer is:", int(num))

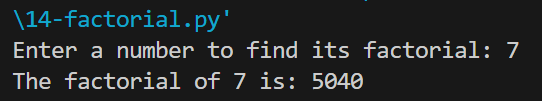
print(f"Length of the word '{word}' is:", len(word))

print(f"The word '{word}' in uppercase is:", word.upper())

print(f"Minimum of digits in [3, 7, 1, 9]:", min(3, 7, 1, 9))

print(f"Maximum of digits in [3, 7, 1, 9]:", max(3, 7, 1, 9))

Output 14 :-



1. **Write a program to implement recursion for factorial of a number that demonstrate the user defined function and return statement.**

def factorial(n):

    if n == 0 or n == 1:

        return 1

    else:

        return n \* factorial(n - 1)

num = int(input("Enter a number to find its factorial: "))

if num < 0:

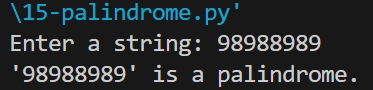
    print("Factorial is not defined for negative numbers.")

else:

    result = factorial(num)

    print(f"The factorial of {num} is: {result}")

Output 15 :-



1. **Write program to check whether the string is palindrome or not.**

def is\_palindrome(s):

    return s == s[::-1]

string = input("Enter a string: ")

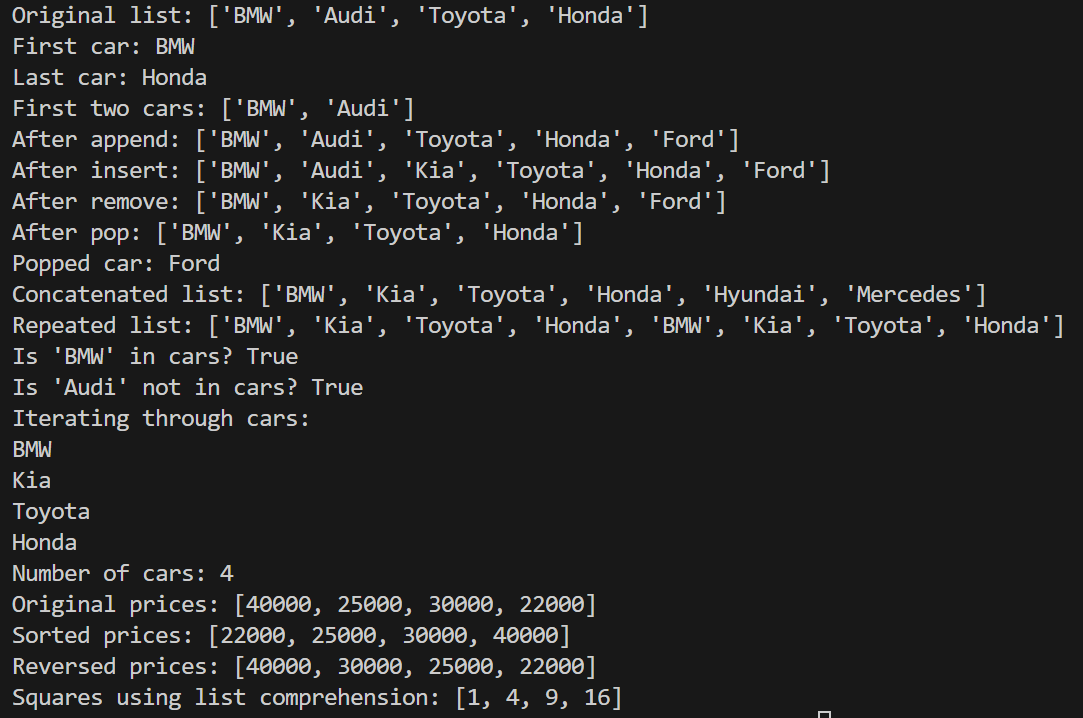
if is\_palindrome(string):

    print(f"'{string}' is a palindrome.")

else:

    print(f"'{string}' is not a palindrome.")

Output 16 :-



1. **Write a program to demonstrate list functions and operations.**

cars = ['BMW', 'Audi', 'Toyota', 'Honda']

print("Original list:", cars)

print("First car:", cars[0])

print("Last car:", cars[-1])

print("First two cars:", cars[0:2])

cars.append('Ford')

print("After append:", cars)

cars.insert(2, 'Kia')

print("After insert:", cars)

cars.remove('Audi')

print("After remove:", cars)

popped\_car = cars.pop()

print("After pop:", cars)

print("Popped car:", popped\_car)

more\_cars = ['Hyundai', 'Mercedes']

all\_cars = cars + more\_cars

print("Concatenated list:", all\_cars)

repeated\_cars = cars \* 2

print("Repeated list:", repeated\_cars)

print("Is 'BMW' in cars?", 'BMW' in cars)

print("Is 'Audi' not in cars?", 'Audi' not in cars)

print("Iterating through cars:")

for car in cars:

    print(car)

print("Number of cars:", len(cars))

prices = [40000, 25000, 30000, 22000]

print("Original prices:", prices)

prices.sort()

print("Sorted prices:", prices)

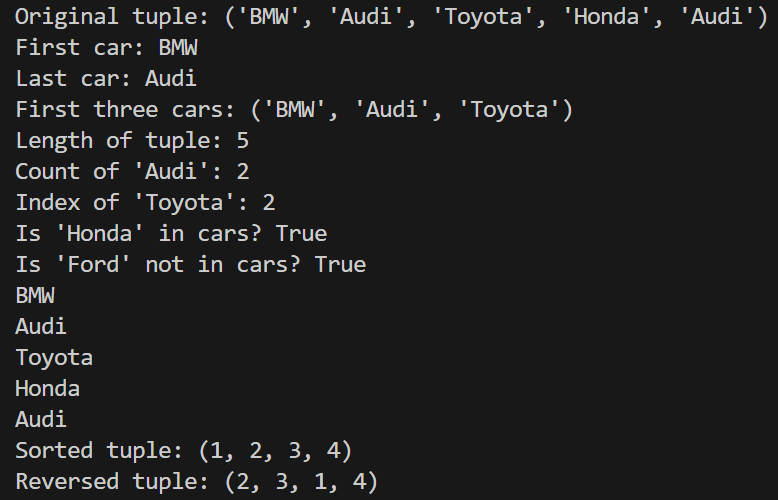
prices.reverse()

print("Reversed prices:", prices)

squared\_prices = [p\*\*2 for p in range(1, 5)]

print("Squares using list comprehension:", squared\_prices)

Output 17 :-



1. **Write a program to demonstrate tuple functions and operations.**

cars = ('BMW', 'Audi', 'Toyota', 'Honda', 'Audi')

print("Original tuple:", cars)

print("First car:", cars[0])

print("Last car:", cars[-1])

print("First three cars:", cars[:3])

print("Length of tuple:", len(cars))

print("Count of 'Audi':", cars.count('Audi'))

print("Index of 'Toyota':", cars.index('Toyota'))

print("Is 'Honda' in cars?", 'Honda' in cars)

print("Is 'Ford' not in cars?", 'Ford' not in cars)

for car in cars:

    print(car)

numbers = (4, 1, 3, 2)

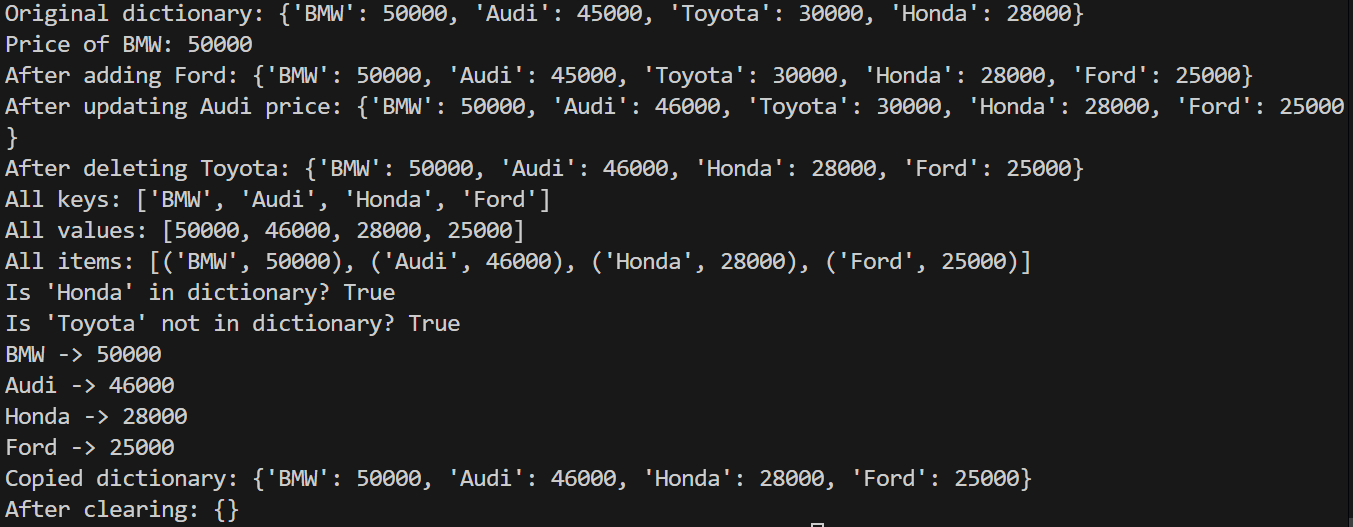
sorted\_numbers = tuple(sorted(numbers))

print("Sorted tuple:", sorted\_numbers)

reversed\_numbers = tuple(reversed(numbers))

print("Reversed tuple:", reversed\_numbers)

Output 18 :-



1. **Write a program to demonstrate dictionary functions and operations.**

car\_prices = {'BMW': 50000, 'Audi': 45000, 'Toyota': 30000, 'Honda': 28000}

print("Original dictionary:", car\_prices)

print("Price of BMW:", car\_prices['BMW'])

car\_prices['Ford'] = 25000

print("After adding Ford:", car\_prices)

car\_prices['Audi'] = 46000

print("After updating Audi price:", car\_prices)

del car\_prices['Toyota']

print("After deleting Toyota:", car\_prices)

print("All keys:", list(car\_prices.keys()))

print("All values:", list(car\_prices.values()))

print("All items:", list(car\_prices.items()))

print("Is 'Honda' in dictionary?", 'Honda' in car\_prices)

print("Is 'Toyota' not in dictionary?", 'Toyota' not in car\_prices)

for car, price in car\_prices.items():

    print(car, "->", price)

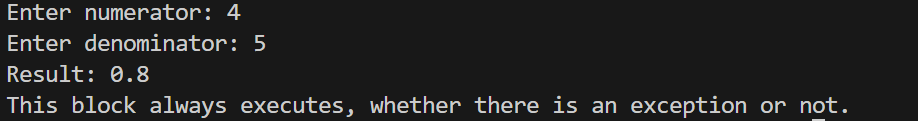
price\_copy = car\_prices.copy()

print("Copied dictionary:", price\_copy)

car\_prices.clear()

print("After clearing:", car\_prices)

Output 19 :-



1. **Write a program to catch a divide by zero exception and add a finally block too.**

def divide\_numbers(a, b):

try:

result = a / b

print(f"Result: {result}")

except ZeroDivisionError:

print("Error: Cannot divide by zero!")

finally:

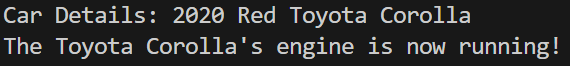
print("This block always executes, whether there is an exception or not.")

num1 = int(input("Enter numerator: "))

num2 = int(input("Enter denominator: "))

divide\_numbers(num1, num2)

Output 20 :-



1. **Write a program to demonstrate classes and their attributes.**

class Car:

    def \_\_init\_\_(self, make, model, year, color):

        self.make = make

        self.model = model

        self.year = year

        self.color = color

    def display\_info(self):

        print(f"Car Details: {self.year} {self.color} {self.make} {self.model}")

    def start\_engine(self):

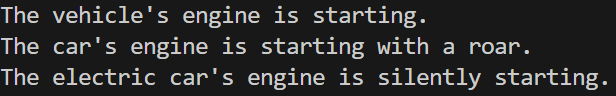
        print(f"The {self.make} {self.model}'s engine is now running!")

car1 = Car("Toyota", "Corolla", 2020, "Red")

car1.display\_info()

car1.start\_engine()

Output 21 :-



1. **Write a program to demonstrate inheritance and method overriding.**

class Car(Vehicle):

    def start\_engine(self):

        print("The car's engine is starting with a roar.")

class ElectricCar(Car):

    def start\_engine(self):

        print("The electric car's engine is silently starting.")

vehicle = Vehicle()

car = Car()

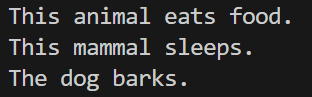
electric\_car = ElectricCar()

vehicle.start\_engine()

car.start\_engine()

electric\_car.start\_engine()

Output 22 :-



1. **Write a program to show the use of multilevel inheritance.**

class Animal:

    def eat(self):

        print("This animal eats food.")

class Mammal(Animal):

    def sleep(self):

        print("This mammal sleeps.")

class Dog(Mammal):

    def bark(self):

        print("The dog barks.")

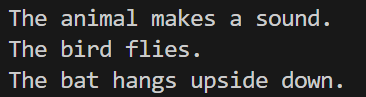
dog = Dog()

dog.eat()

dog.sleep()

dog.bark()

Output 23 :-



1. **Write a program to show the use of multiple inheritance.**

class Animal:

    def speak(self):

        print("The animal makes a sound.")

class Bird:

    def fly(self):

        print("The bird flies.")

class Bat(Animal, Bird):

    def hang(self):

        print("The bat hangs upside down.")

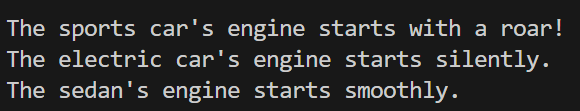
bat = Bat()

bat.speak()

bat.fly()

bat.hang()

Output 24 :-



1. **Write a program to implement polymorphism**

class Car:

    def start\_engine(self):

        print("The car's engine is starting.")

class SportsCar(Car):

    def start\_engine(self):

        print("The sports car's engine starts with a roar!")

class ElectricCar(Car):

    def start\_engine(self):

        print("The electric car's engine starts silently.")

class Sedan(Car):

    def start\_engine(self):

        print("The sedan's engine starts smoothly.")

def start\_car\_engine(car):

    car.start\_engine()

sports\_car = SportsCar()

electric\_car = ElectricCar()

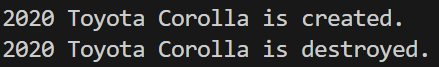
sedan = Sedan()

start\_car\_engine(sports\_car)

start\_car\_engine(electric\_car)

start\_car\_engine(sedan)

Output 25 :-



1. **Write a program to implement constructor and destructor.**

class Car:

    def \_\_init\_\_(self, make, model, year):

        self.make = make

        self.model = model

        self.year = year

        print(f"{self.year} {self.make} {self.model} is created.")

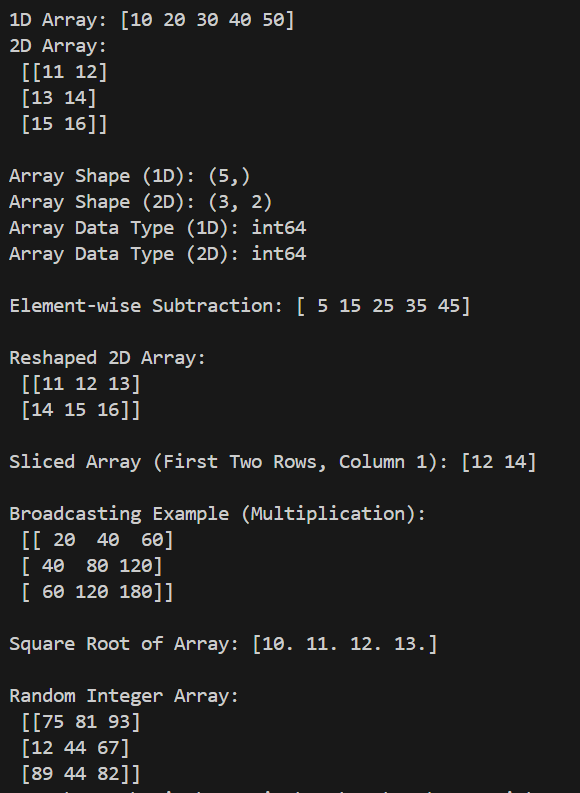
    def \_\_del\_\_(self):

        print(f"{self.year} {self.make} {self.model} is destroyed.")

car1 = Car("Toyota", "Corolla", 2020)

del car1

Output 26 :-



1. **Write a program to demonstrate NumPy basics**

import numpy as np

# 1. Creating NumPy arrays

arr1 = np.array([10, 20, 30, 40, 50])

arr2 = np.array([[11, 12], [13, 14], [15, 16]])

print("1D Array:", arr1)

print("2D Array:\n", arr2)

# 2. Array properties

print("\nArray Shape (1D):", arr1.shape)

print("Array Shape (2D):", arr2.shape)

print("Array Data Type (1D):", arr1.dtype)

print("Array Data Type (2D):", arr2.dtype)

# 3. Array Operations

arr3 = np.array([5, 5, 5, 5, 5])

difference\_arr = arr1 - arr3

print("\nElement-wise Subtraction:", difference\_arr)

# 4. Array Reshaping

reshaped\_arr = arr2.reshape(2, 3)

print("\nReshaped 2D Array:\n", reshaped\_arr)

# 5. Array Slicing

sliced\_arr = arr2[0:2, 1]

print("\nSliced Array (First Two Rows, Column 1):", sliced\_arr)

# 6. Array Broadcasting

arr4 = np.array([10, 20, 30])

arr5 = np.array([[2], [4], [6]])

broadcasted\_arr = arr4 \* arr5

print("\nBroadcasting Example (Multiplication):\n", broadcasted\_arr)

# 7. Array Mathematical Functions

arr6 = np.array([100, 121, 144, 169])

sqrt\_arr = np.sqrt(arr6)

print("\nSquare Root of Array:", sqrt\_arr)

# 8. Random Array

random\_arr = np.random.randint(1, 100, size=(3, 3))

print("\nRandom Integer Array:\n", random\_arr)