**1.Write a python program to calculate the area of a rectangle given its length and width?**

Ans.

Program :

def calculate\_rectangle\_area(length, width):

area = length \* width

return area

def main():

length = float(input("Enter the length of the rectangle: "))

width = float(input("Enter the width of the rectangle: "))

area = calculate\_rectangle\_area(length, width)

print("The area of the rectangle is:", area)

if \_name\_ == "\_main\_":

    main()

**2.write a program to convert miles to kilometers?**

Ans:

Program:

def miles\_to\_kilometers(miles):

kilometers = miles \* 1.60934

return kilometers

def main():

miles = float(input("Enter the distance in miles: "))

kilometers = miles\_to\_kilometers(miles)

print(f"{miles} miles is equal to {kilometers:.2f} kilometers.")

if \_name\_ == "\_main\_":

    main()

**3.Write a function to check if a given string is a palindrome?**

Ans:

Program:

def is\_palindrome(s):

s = s.lower().replace(" ", "")

return s == s[::-1]

def main():

user\_input = input("Enter a string: ")

if is\_palindrome(user\_input):

print("Yes, it's a palindrome.")

else:

print("No, it's not a palindrome.")

if \_name\_ == "\_main\_":

    main()

**4.Write a python program to find the second largest element in a list?**

Ans:

Program:

def find\_second\_largest(arr):

if len(arr) < 2:

return "List should have at least two elements"

largest = max(arr[0], arr[1])

second\_largest = min(arr[0], arr[1])

for i in range(2, len(arr)):

if arr[i] > largest:

second\_largest = largest

largest = arr[i]

elif arr[i] > second\_largest and arr[i] != largest:

second\_largest = arr[i]

return second\_largest

def main():

arr = [int(x) for x in input("Enter elements of the list separated by spaces: ").split()]

second\_largest = find\_second\_largest(arr)

print("The second largest element in the list is:", second\_largest)

if \_name\_ == "\_main\_":

    main()

**5. Explain what indentation means in python?**

**Ans:**

In Python, indentation refers to the whitespace at the beginning of a line of code. Unlike many other programming languages, such as C or Java, which use curly braces {} to define blocks of code, Python uses indentation to delimit blocks of code.

Indentation is crucial in Python because it determines the structure of the code and is used to indicate which statements are part of a particular block or control structure, such as loops, conditionals, functions, or classes.

Here's an example to illustrate how indentation works in Python:

if condition:

print("Condition is True")

else:

print("Condition is False")

The lines of code following the if and else statements are indented to indicate that they belong to their respective blocks of code.

The indentation level is typically four spaces or a tab character, but the most important thing is to be consistent with whichever you choose.

Incorrect indentation will lead to syntax errors or incorrect program behavior.

Python's use of indentation promotes clean and readable code, but it can sometimes be a source of frustration for programmers coming from other languages where indentation is not as significant.

**6.write a program to perform set difference operation?**

Program:

def set\_difference\_using\_operator(set1, set2):

return set1 - set2

def set\_difference\_using\_method(set1, set2):

return set1.difference(set2)

def main():

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

set2 = {3, 4, 5, 6, 7}

difference\_operator = set\_difference\_using\_operator(set1, set2)

difference\_method = set\_difference\_using\_method(set1, set2)

print("Set difference using operator (-):", difference\_operator)

print("Set difference using method (.difference()):", difference\_method)

if \_name\_ == "\_main\_":

    main()

**7.Write a python program to print numbers from 1 to 10 using a while loop?**

Ans:

num = 1

while num <= 10:

print(num)

num += 1

This program initializes a variable num with the value 1. Then, it enters a while loop that continues as long as num is less than or equal to 10. Inside the loop, the current value of num is printed, and then num is incremented by 1 in each iteration. This loop continues until num reaches 10, printing the numbers from 1 to 10 along the way.

**8.write a program to calculate the factorial of a numbers using a while loop?**

**Ans:**

def factorial(n)

result = 1

while n > 0:

result \*= n

n -= 1

return result

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

if number < 0:

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

print("Factorial of", number, "is:", fact)

**9.Write a python program to check if a number is positive, negative or zero using if elif else statement?**

Ans:

Input number from the user

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

if number > 0:

print("The number is positive.")

elif number < 0:

print("The number is negative.")

else:

print("The number is zero.")

In this program:

We take the input number from the user using the input() function.

The input is converted to a float to handle decimal numbers as well.

We use if-elif-else statements to check whether the number is positive, negative, or zero.

If the number is greater than 0, it's considered positive.

If the number is less than 0, it's considered negative.

If the number is neither greater nor less than 0, it's considered zero.

The appropriate message is printed based on the condition met.

**10.Write a program to determine the largest among three number using conditional statement?**

Ans:

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

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

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

if num1 >= num2 and num1 >= num3:

largest = num1

elif num2 >= num1 and num2 >= num3:

largest = num2

else:

largest = num3

print("The largest number among", num1, ",", num2, ", and", num3, "is:", largest)

**11.Write a python program to create numpy array filled with ones of given shape?**

Ans:

import numpy as np

def create\_ones\_array(shape):

"""

Create a NumPy array filled with ones of the given shape.

Parameters:

shape (tuple): Shape of the array.

Returns:

numpy.ndarray: Array filled with ones.

"""

return np.ones(shape)

shape = (3, 4) # Shape of the array

ones\_array = create\_ones\_array(shape)

print("Array filled with ones:")

print(ones\_array)

**12.Write a program to create a 2D numpy array initialized with random integers?**

Ans:

import numpy as np

def create\_random\_array(rows, cols, min\_val, max\_val):

"""

Create a 2D NumPy array initialized with random integers.

Parameters:

rows (int): Number of rows in the array.

cols (int): Number of columns in the array.

min\_val (int): Minimum value for random integers.

max\_val (int): Maximum value for random integers.

Returns:

numpy.ndarray: 2D array initialized with random integers.

"""

return np.random.randint(min\_val, max\_val + 1, size=(rows, cols))

rows = 3

cols = 4

min\_val = 0

max\_val = 10

random\_array = create\_random\_array(rows, cols, min\_val, max\_val)

print("Random 2D array:")

print(random\_array)

**13.Write a python program to generate an array of evenly spaced numbers over a specified range using linspace?**

Ans:

import numpy as np

def generate\_linspace(start, stop, num):

"""

Generate an array of evenly spaced numbers over a specified range using linspace.

Parameters:

start (float): Start of the range.

stop (float): End of the range.

num (int): Number of samples to generate.

Returns:

numpy.ndarray: Array of evenly spaced numbers.

"""

return np.linspace(start, stop, num)

start = 0

stop = 10

num = 20

linspace\_array = generate\_linspace(start, stop, num)

print("Array of evenly spaced numbers over the range [{}, {}]:".format(start, stop))

print(linspace\_array)

**14.Write a program to generate an array of 10 equally spaced values between 1 and 100 using linspace?**

Ans:

import numpy as np

# Generate an array of 10 equally spaced values between 1 and 100 using linspace

equally\_spaced\_values = np.linspace(1, 100, 10)

print("Array of 10 equally spaced values between 1 and 100:")

print(equally\_spaced\_values)

In this program:

We import the NumPy library as np.

We use NumPy's np.linspace() function to generate an array of 10 equally spaced values between 1 and 100.

The linspace() function takes three arguments: the start of the range (1), the end of the range (100), and the number of samples (10).

The function generates 10 equally spaced values between 1 and 100 inclusively.

Finally, we print the generated array of 10 equally spaced values.

**15.Write a python program to create an array containing even numbers from 2 to 20 using arrange?**

Ans:

import numpy as np

# Create an array containing even numbers from 2 to 20 using arange

even\_numbers = np.arange(2, 21, 2)

# Print the generated array

print("Array containing even numbers from 2 to 20:")

print(even\_numbers)

In this program:

We import the NumPy library as np.

We use NumPy's np.arange() function to create an array containing even numbers from 2 to 20.

The arange() function takes three arguments: the start of the range (2), the end of the range (21 excluding), and the step size (2).

The function generates numbers starting from 2 (inclusive) up to 21 (exclusive) with a step size of 2, resulting in even numbers.

Finally, we print the generated array containing even numbers from 2 to 20.

**16.Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using arrange?**

Ans:

import numpy as np

# Create an array containing numbers from 1 to 10 with a step size of 0.5 using arange

numbers = np.arange(1, 10.5, 0.5)

# Print the generated array

print("Array containing numbers from 1 to 10 with a step size of 0.5:")

print(numbers)

In this program:

We import the NumPy library as np.

We use NumPy's np.arange() function to create an array containing numbers from 1 to 10 with a step size of 0.5.

The arange() function takes three arguments: the start of the range (1), the end of the range (10.5 excluding), and the step size (0.5).

The function generates numbers starting from 1 (inclusive) up to 10.5 (exclusive) with a step size of 0.5.

Finally, we print the generated array containing numbers from 1 to 10 with a step size of 0.5.