Date: February 14, 2024

SMART INTERNZ - APSCHE

AI / ML Training

Assessment

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

```
def calculate_rectangle_area(length, width):
    return length * width
def main():
    length, width = map(float, input("Enter length and width of the rectangle separated by space: ").split())
    area = calculate_rectangle_area(length, width)
    print("The area of the rectangle is:", area)
if __name__ == "__main__":
    main()
```

```
input
Enter length and width of the rectangle separated by space: 24 22
The area of the rectangle is: 528.0
```

2. Write a program to convert miles to kilometers

```
main.py
miles = float(input("Enter distance in miles: "))
kilometers = miles * 1.60934
print(f"{miles} miles is equal to {kilometers} kilometers.")
```

```
input
Enter distance in miles: 10
10.0 miles is equal to 16.0934 kilometers.
```

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

```
main.py
def is_palindrome(s):
    s = ''.join(char.lower() for char in s if char.isalnum())
    return s == s[::-1]
if __name__ == "__main__":
    test_string = input("Enter a string to check if it's a palindrome: ")
    if is_palindrome(test_string):
        print("Yes, it's a palindrome.")
    else:
        print("No, it's not a palindrome.")
```

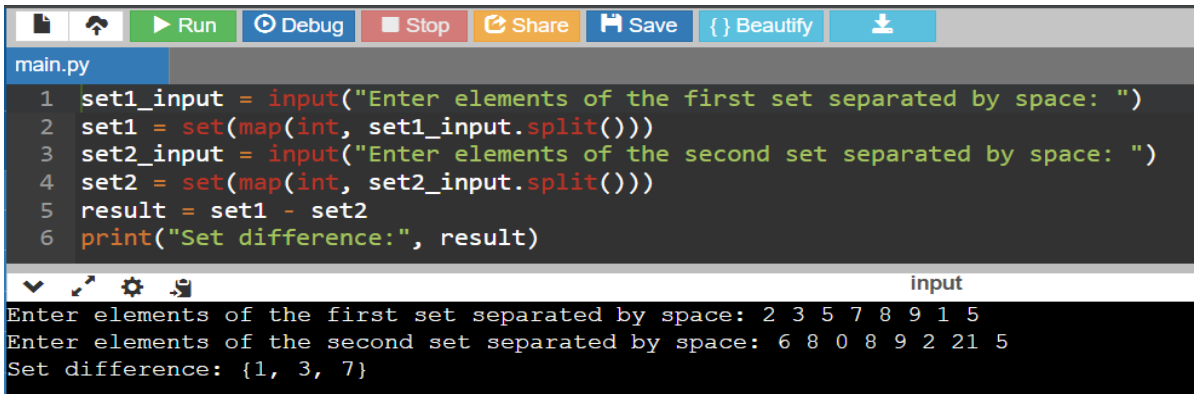
```
input
Enter a string to check if it's a palindrome: 7
Yes, it's a palindrome.
```

4. Write a Python program to find the second largest element in a list.

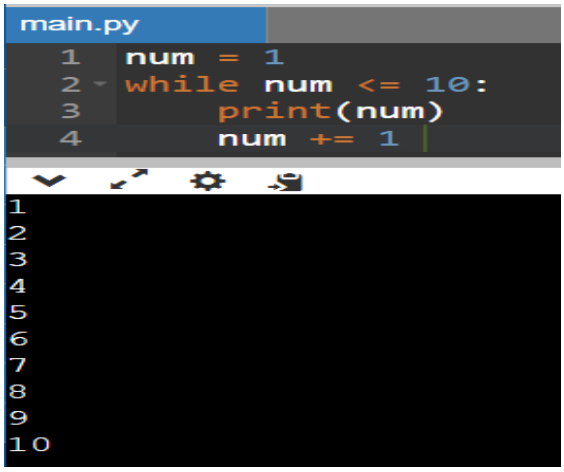
```
main.py
nums = [int(x) for x in input("Enter elements of the list separated by space: ").split()]
nums.sort()
print("The second largest element in the list is:", nums[-2])
```

```
input
Enter elements of the list separated by space: 23 45 67 89 98 654 34 46 32 33 22 28 67
The second largest element in the list is: 98
```

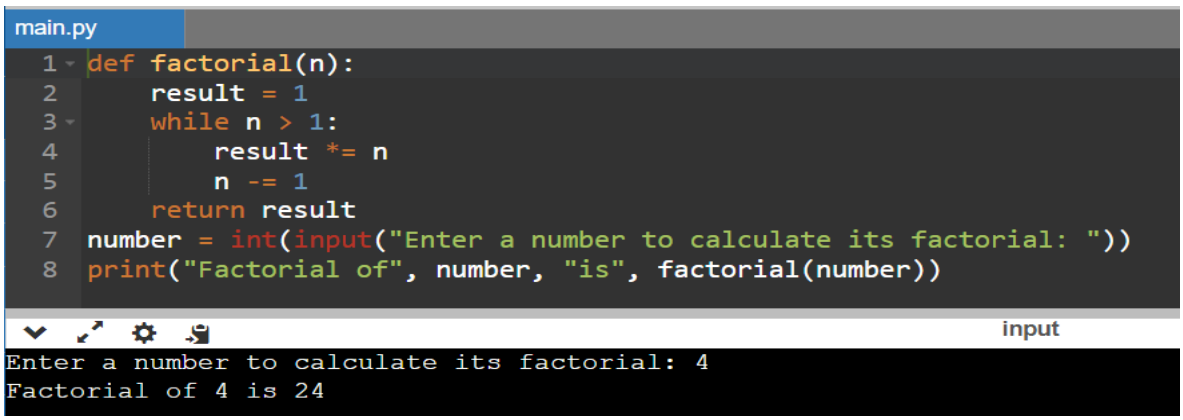
5. Explain what indentation means in Python.
    - Indentation in Python determines the structure of code blocks.
    - Blocks of code, such as loops and conditionals, are defined by indentation.
    - Consistent indentation, typically 4 spaces or a tab, is crucial for Python's syntax.
    - Python does not use braces to denote code blocks; indentation serves this purpose.
    - Proper indentation enhances code readability and understanding of program structure.
6. Write a program to perform set difference operation.



7. Write a Python program to print numbers from 1 to 10 using a while loop.



8. Write a program to calculate the factorial of a number using a while loop.



9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements.

```python
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.")
```

```
Enter a number: 10
The number is positive.
```

10. Write a program to determine the largest among three numbers using conditional statements.

```python
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the 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 is:", largest)
```

```
Enter the first number: 4
Enter the second number: 6
Enter the third number: 10
The largest number is: 10.0
```

11. Write a Python program to create a numpy array filled with ones of given shape.

main.py

```python
import numpy as np
def create_ones_array(shape):
    ones_array = np.ones(shape)
    return ones_array
shape = tuple(map(int, input("Enter the shape of the array (separated by space): ").split()))
ones_array = create_ones_array(shape)
print("Numpy array filled with ones:")
print(ones_array)
```

input

```
Enter the shape of the array (separated by space): 3 4
Numpy array filled with ones:
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

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

```python
import numpy as np
def create_random_array(rows, cols):
    random_array = np.random.randint(low=0, high=100, size=(rows, cols))
    return random_array
rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))
random_array = create_random_array(rows, cols)
print("2D numpy array initialized with random integers:")
print(random_array)
```

input

```
Enter the number of rows: 3
Enter the number of columns: 4
2D numpy array initialized with random integers:
[[86 30 87 13]
 [48 76 86  2]
 [97 40 59 43]]
```

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

```
import numpy as np
def generate_linspace_array(start, stop, num):
    linspace_array = np.linspace(start, stop, num)
    return linspace_array
start = float(input("Enter the start value: "))
stop = float(input("Enter the stop value: "))
num = int(input("Enter the number of elements: "))
linspace_array = generate_linspace_array(start, stop, num)
print("Array of evenly spaced numbers over the specified range:")
print(linspace_array)
```

```
Enter the start value: 5
Enter the stop value: 27
Enter the number of elements: 10
Array of evenly spaced numbers over the specified range:
[ 5.          7.44444444  9.88888889 12.33333333 14.77777778 17.22222222
 19.66666667 22.11111111 24.55555556 27.        ]
```

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

```
import numpy as np
linspace_array = np.linspace(1, 100, 10)
print("Array of 10 equally spaced values between 1 and 100:")
print(linspace_array)
```

```
Array of 10 equally spaced values between 1 and 100:
[  1.  12.  23.  34.  45.  56.  67.  78.  89. 100.]
```

15. Write a Python program to create an array containing even numbers from 2 to 20 using arange.

```
import numpy as np
even_numbers_array = np.arange(2, 21, 2)
print("Array containing even numbers from 2 to 20:")
print(even_numbers_array)
```

```
Array containing even numbers from 2 to 20:
[ 2  4  6  8 10 12 14 16 18 20]
```

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

```
import numpy as np
numbers_array = np.arange(1, 10.1, 0.5)
print("Array containing numbers from 1 to 10 with a step size of 0.5:")
print(numbers_array)
```

```
Array containing numbers from 1 to 10 with a step size of 0.5:
[ 1.   1.5  2.   2.5  3.   3.5  4.   4.5  5.   5.5  6.   6.5  7.   7.5
  8.   8.5  9.   9.5 10. ]
```