Assignment - 2

Q:1-Accept a floating-point number from the user and use the built-in functions round(), int(), and abs() to display its rounded value, integer value, and absolute value respectively.
Explain the difference between round() and int() in the output. Second, take a list of integers as input and use the max(), min(), and sum() built-in functions to find the largest, smallest, and total of the numbers. Display each result with a clear label.

```
# Accept a floating-point number
num = float(input("Enter a floating-point number: "))

# Using built-in functions
print("Rounded value:", round(num))
print("Integer value:", int(num))
print("Absolute value:", abs(num))

# Explanation
# round() rounds the number to the nearest integer (example: 4.6 -> 5, 4.4
-> 4)
# int() just removes the decimal part without rounding (example: 4.9 -> 4)
# abs() gives the positive version of the number

# Accepting a list of integers
nums = list(map(int, input("Enter a list of integers separated by space:
").split()))

# Using max, min, sum
print("Largest number:", max(nums))
print("Smallest number:", min(nums))
print("Sum of all numbers:", sum(nums))
```

Output:

```
Enter a floating-point number: 4.6
Rounded value: 5
Integer value: 4
Absolute value: 4.6
Enter a list of integers separated by space: 4 6
Largest number: 6
Smallest number: 4
Sum of all numbers: 10
```

Q:2-Create a program that takes a string input and uses the built-in len(), sorted(), and type() functions to display the string's length, its sorted character list, and the data type of the result.

```
text = input("Enter a string: ")

print("Length of string:", len(text))
print("Sorted characters:", sorted(text))
print("Type of sorted result:", type(sorted(text)))
```

Output:

```
Enter a string: hasti
Length of string: 5
Sorted characters: ['a', 'h', 'i', 's', 't']
Type of sorted result: <class 'list'>
```

Q:3- Accept two numbers from the user and use the pow() function to compute the result of raising the first number to the power of the second. Then compare it with using the ** operator and explain whether they behave the same or differently.

```
a = int(input("Enter the base number: "))
b = int(input("Enter the exponent: "))

print("Using pow() result:", pow(a, b))
print("Using ** operator result:", a ** b)

# Both pow(a, b) and a**b give the same result for positive integers.
```

Output:

```
Enter the base number: 2
Enter the exponent: 3
Using pow() result: 8
Using ** operator result: 8
```

Q:4- Ask the user to input a string and a substring. Use the .find() and .count() methods to check whether the substring exists in the original string, its index position, and how many times it appears.

```
main_str = input("Enter the main string: ")
sub_str = input("Enter the substring to find: ")

index = main_str.find(sub_str)
count = main_str.count(sub_str)

if index != -1:
    print(f"Substring found at index {index}")
    print(f"Substring appears {count} times")
else:
    print("Substring not found")
```

Output :

```
Enter the main string: wordhunt
Enter the substring to find: hunt
Substring found at index 4
Substring appears 1 times
```

Q:5- Write a Python program that takes a string input from the user and uses slicing to extract and display the first 5 characters, the last 5 characters, and the characters from position 3 to position 8 (excluding 8).

```
s = input("Enter a string: ")

print("First 5 characters:", s[:5])
print("Last 5 characters:", s[-5:])
print("Characters from position 3 to 8:", s[3:8])
```

Output :

```
Enter a string: hasti
First 5 characters: hasti
Last 5 characters: hasti
Characters from position 3 to 8: ti
```

Q:6- Write a program that asks the user to input a string and displays the string with only alternate characters in reverse order using slicing. (For example, "abcdef" should give "fdb").

```
s = input("Enter a string: ")
# Reverse and take alternate characters
print("Alternate characters in reverse order:", s[::-2])
```

Output:

```
Enter a string: Hasti
Alternate characters in reverse order: isH
```

Q:7- Given a string, use slicing to print:

- **The string with the middle character removed**
- **The string in reverse with every second character**

And take a string input and extract a substring using slicing from index 2 to 8. Then, extract another substring starting from the middle of the string to the end. Display both substrings.

```
s = input("Enter a string: ")
# Remove middle character
middle_index = len(s) // 2
if len(s) % 2 == 0:
    removed_middle = s
else:
    removed_middle = s[:middle_index] + s[middle_index+1:]
print("Without middle character:", removed_middle)
# Reverse with every second character
print("Reverse string with every second character:", s[::-2])
# Substring from index 2 to 8
print("Substring from index 2 to 8:", s[2:8])
# Substring from middle to end
print("Substring from middle to end:", s[len(s)//2:])
```

Output:

```
Enter a string: hasti
Without middle character: hati
Reverse string with every second character: ish
Substring from index 2 to 8: sti
Substring from middle to end: sti
```

Q:8- Accept a string and use slicing to swap the first three and last three characters. Print the resulting modified string. And write a Python program that takes a string and removes the first and last characters using slicing. Then print the shortened string.

```
s = input("Enter a string: ")

# Swap first 3 and last 3
if len(s) >= 6:
    new_str = s[-3:] + s[3:-3] + s[:3]
    print("Swapped string:", new_str)
else:
    print("String too short to swap 3 characters")

# Remove first and last character
print("Without first and last character:", s[1:-1])
```

Output:

```
Enter a string: hasti
String too short to swap 3 characters
Without first and last character: ast
```

Q:9- Create a calculator that takes user input (with interactive python window) and performs all arithmetic and logical operations. (Don't use any in-built functions)

```
print("Simple Calculator")
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b if b != 0 else "Cannot divide by zero")
print("Modulus:", a % b)
print("Power:", a ** b)

# Logical operations
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)
```

Output:

```
Simple Calculator
Enter first number: 20
Enter second number: 10
Addition: 30
Subtraction: 10
Multiplication: 200
Division: 2.0
Modulus: 0
Power: 10240000000000
a == b: False
a != b: True
a > b: True
a < b: False
a >= b: True
a <= b: False
```

***