Lab 3

Factorial Calculator (10 pts)

Write a Python program that computes the factorial of a positive integer.

What is a Factorial?
The factorial of a number n, written as **n!**, is the product of all positive integers from **1** up to **n**:

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

Examples:

- $4! = 4 \times 3 \times 2 \times 1 = 24$
- $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Requirements:

1. Ask the user for a positive integer.
2. Validate input: keep asking if it’s not a positive integer or not an integer.
3. Use a loop to multiply numbers from 1 up to the input value.

Example:

```
Enter a positive integer: 5
5! = 120
```

Inverse Digit (20 pts)

Write a Python program that reverses the digits of a positive integer.

Requirements:

1. The program should use `input()` to ask the user for a positive integer.
2. Validate the input: If the user enters a negative number or something that is not a whole number, keep asking until a valid positive integer is provided.
3. This question can be done using `while` loops.

Example:

```
Enter a positive integer: 12340
Reversed number: 4321
```

Remove Consecutive duplicates (20 pts)

Write a Python program that removes consecutive duplicate characters from a given string `S`.

Requirements:

1. The program should use `input()` to ask the user for a string `S`.
2. Define a new string and update it in the loop.

Examples:

```
Input: "aabbcCddeee"
Output: "abcCde"
```

```
Input: "helloo wooorld"
Output: "helo world"
```

```
Input: "aaabb@@@aaccd"
Output: "ab@acd
```

Hint:

- Use `+` add the i-th character of `S` to another string `A`:

  ```
  A = A + S[i]
  ```

Guess Number (20 pts)

Write a Python program that plays a guessing game where the computer tries to guess the user's secret number between 1 and 100. It uses a binary search strategy, and the user responds with "higher," "lower," or "correct".

Requirements

1. Prompt the user to think of a number between 1 and 100 (User shouldn't input the number).
2. The computer prints its guess. And use `input()` to the user for responses:
    - The responses should be one of “higher,” “lower,” and “correct”
    - Keep asking until a valid response is given.
3. Do step 2 iteratively, until the user responds with “correct”

Examples:

```
Think of a number between 1 and 100.
Is your number 50? higher
Is your number 75? lower
Is your number 62? higher
Is your number 68? correct
The computer guessed your number in 4 tries!
```