I/O Practice Problems

1. Simple Greeting

Write a Python program that:

- Asks the user for their name using `input()`
- Prints a greeting like: `Hello Alice`

Questions:

- What data type does `input()` return for the name?
- How could you check this inside your program?

2. Name and Age Sentence

Write a Python program that:

- Asks for the user's name
- Asks for the user's age
- Prints: `Hello <name>, you are <age> years old.`

Then:

- Print out the **type** of the `age` variable.
- Explain in a comment why that type might be a problem if you want to do maths with age.

3. Age Next Year (Type Conversion)

Start from your solution to Problem 2.

- Modify the program so that `age` is stored as an `int`
- Calculate `age_next_year = age + 1`
- Print: `Next year you will be <age_next_year>`

If you forget to convert to `int`, what error do you get?

4. Simple Two-Number Calculator

Write a program that:

- Asks the user to enter two numbers
- Stores them as `float`
- Calculates their **sum** and **product**
- Prints both results with clear messages.

Try it with:

- Two integers (e.g. 2 and 3)
- Two decimals (e.g. 2.5 and 3.1)

What data type do you see for each result?

5. Dynamic Typing with `x`

Write a Python program that:

- Sets `x = 10`
- Prints `x` and `type(x)`
- Then sets `x = "Hello"`
- Prints `x` and `type(x)` again

Then, try to run `x + 1` after `x = "Hello"`. What happens, and why?

6. Fix the Type Confusion Bug

You are given this buggy program:

```
age = input("Enter age: ")
age_next_year = age + 1
print(age_next_year)
```

Tasks:

1. Predict what will happen when you run it.
2. Run it and note the exact error message.
3. Fix the program so it works correctly and prints `Next year you will be <age_next_year>`.

7. Naming Variables – Make It Clear

You are given this Python code:

```
a = 50
b = 60
c = a + b
print(c)
```

Tasks:

1. Decide on a *meaning* for this code (e.g. total marks for two tests).
2. Rename the variables to be more descriptive.
3. Rewrite the code using **lowercase_with_underscores**.

Example starting point: think of `a` as “test 1 mark”.

8. User Profile Program

Write a program that:

- Asks for `name` (string)

- Asks for `age` (int)
- Asks for `height` in metres (float)
- Calculates `age_plus_one = age + 1`
- Prints:
    - `Hello <name>`
    - `Next year you will be <age_plus_one>`
    - `Your height is <height> metres`

Then:

- Print the types of `age` and `height` to confirm they're numeric.

9. Product Cost Calculator

Write a program that:

- Asks for `product_name` (string)
- Asks for `price` (float)
- Asks for `quantity` (int)
- Calculates `total_cost = price * quantity`
- Prints:
    - `You bought <quantity> of <product_name>`
    - `Total cost is <total_cost>`

Make sure you choose the **correct data type** for each variable.

10. Spot the Problems (Mixed Concepts)

Consider the following code:

```
Name = input("Enter your name: ")
AGE = input("Enter your age: ")
age_next_year = AGE + 1
print("hello", Name, "next year you will be", age_next_year)
```

Tasks:

1. List at least **three** issues with:
    - Data types
    - Naming style
    - The calculation
2. Rewrite the program so that:

- It follows the naming rules from the lesson
- It runs without errors
- It prints a clear, correct message about the age next year.