Assignment for Exception Handling

Q1: Basic Exception Handling

Write a program that:

1. Accepts two numbers as input from the user.
2. Divides the first number by the second.
3. Catches the following exceptions:
    - Division by zero.
    - Invalid input (non-numeric input).

Hint: Use `try`, `except`, and `finally` blocks.

Q2: Handling Multiple Exceptions

Write a program that:

1. Accepts a filename from the user.
2. Tries to open the file and read its content.
3. Handles the following exceptions:
    - File not found.
    - Permission error.

Q3: Custom Exception

Create a custom exception called `AgeTooSmallError`. Write a program that:

1. Accepts a user's age as input.
2. Raises the `AgeTooSmallError` if the entered age is below 18, with an appropriate error message.

Hint: Use `raise` to trigger the exception.

Q4: Nested Exception Handling

Write a program that:

1. Accepts a list of numbers from the user (comma-separated).

2. Converts them into integers.
3. Prints the sum of all numbers.
4. Handles the following exceptions:
    - Invalid format (non-integer inputs in the list).
    - Empty input.
5. Uses nested `try` blocks to handle different exceptions.

Q5: Exception Propagation

Write a program with two functions:

1. A function `calculate_area()` that calculates the area of a rectangle given its `length` and `width` as arguments. If either value is negative, it raises a `ValueError`.
2. A function `main()` that calls `calculate_area()` and handles the exception gracefully.

Q6: Else Block in Exception Handling

Write a program that:

1. Prompts the user to enter a number.
2. Checks if the number is even or odd.
3. Uses the `else` block to print a success message if no exceptions occur.