Assignment: Files, Modules, and Packages

Part 1: File Handling

1. **Reading a File**
   - Write a Python program to read a text file named `students.txt` and print each line with its line number.
2. **Writing to a File**
   - Create a program that writes a list of 5 motivational quotes to a file named `quotes.txt`. Each quote should be written on a new line.
3. **Appending to a File**
   - Add 3 more quotes to the same `quotes.txt` file without overwriting the existing content.
4. **Counting Words in a File**
   - Write a program that reads `quotes.txt` and counts the total number of words in the file.

Part 2: Modules

5. **Using Built-in Modules**
   - Use the `math` module to calculate the square root, factorial, and logarithm of a number entered by the user.
6. **Creating a Custom Module**
   - Create a module named `calculator.py` with the following functions:
     - `add(a, b)`
     - `subtract(a, b)`
     - `multiply(a, b)`
     - `divide(a, b)`
   - Import the module into a separate Python file and use each function to perform calculations.
7. **Random Module**
   - Write a program that generates a random number between 1 and 50 until the number 25 is generated. Print each random number and stop when 25 is found.

Part 3: Packages

8. **Creating a Package**

Create a package named `utilities` with the following structure:
markdown
CopyEdit

```
utilities/
    __init__.py
    string_utils.py
    math_utils.py
```

    -
    - `string_utils.py` should contain a function to reverse a string.
    - `math_utils.py` should contain a function to calculate the power of a number.
    - Write a program to import and use both functions from the package.
9. **Using a Built-in Package**
    - Use the `os` module to list all the files in the current directory and save their names into a file named `file_list.txt`.

Part 4: Integration Task

10. **Combining Files and Modules**
    - Write a program that does the following:
        1. Reads numbers from a file named `numbers.txt`.
        2. Uses a custom module named `stats.py` to calculate the average and maximum of the numbers.
        3. Writes the results (average and maximum) to a file named `results.txt`.

Submission Guidelines:

- Ensure your code is properly commented and easy to read.
- Use meaningful variable names and follow Python coding conventions.
- Test your programs before submission to ensure correctness.
- Save your code in separate Python files for each question.
- Submit your work screen shots with proper file names for each task.