Write a shell script that performs the tasks below. Assume your current working directory is **base_dir**.

1. Prompt the user for a directory name and assign it to a variable `directory_name`.
    a. If the directory exists, proceed to the next step.
    b. If not, print `'No such directory. Creating new.'` and create it.
    c. *Hint*: You can use `-d $directory_name` inside an `if` condition.
2. Change into the given directory. Prompt the user for a **username** (e.g., `root`, `student`, `user`) and store it into a variable called `user_name`.
3. Use a `for` loop to iterate through all `.txt` files in the directory and its subdirectories that are owned by the given `user_name`.
    a. For each such file, output the following: i) File name, ii) Character count, iii) Size class
    b. Your output should be written to a file named `print.log` under the **base_dir**.
4. The size class is defined as follows:
    a. **Empty**: If the character count is zero.
    b. **Small**: If the character count is between 1 and 250 (inclusive).
    c. **Big**: If the file contains more than 250 characters.

Write a shell script that performs the tasks below. Assume your current working directory is **base_dir**.

1. Prompt the user for a directory name and assign it to a variable `directory_name`.
    a. If the directory exists, proceed to the next step.
    b. If not, print `'No such directory. Creating new.'` and create it.
    c. *Hint*: You can use `-d $directory_name` inside an `if` condition.
2. Change into the given directory. Prompt the user for a **file extension** (e.g., `txt`, `log`, `sh`) and store it into a variable called `file_ext`.
3. Use a `for` loop to iterate through all files in the directory and its subdirectories that match the given extension and are writable by the current user.
    a. For each such file, print the following: i) File name, ii) Word count, iii) Size class
    b. Your output should be written to a file named `print.log` under the **base_dir**.
    c. *Hint: You can use* `-w $file_name` *inside an if condition to see if the file is*

writable.

4. The size class is defined as follows:
    a. **Empty**: If the word count is zero.
    b. **Small**: If the word count is between 1 and 50 (inclusive).
    c. **Big**: If the file contains more than 50 words.

Write a shell script that performs the following tasks:

1. Prompt the user for a directory name and assign it to a variable `directory_name`.
    - If the directory exists, proceed to the next step.
    - If not, print `'No such directory'` and exit.
    - *Hint*: You can use `-d $directory_name` inside an `if` condition.

2. Change into the given directory. Prompt the user for a **username** (e.g., `root`, `student`, `user`) and store it into a variable called `user_name`.

3. Use a `for` loop to iterate through all `.txt` files in the directory and its subdirectories that are owned by the given `user_name`.
    - For each such file, print the following: i) File name, ii) Word count, iii) Size class

4. The size class is defined as follows:
    - **Empty**: If the word count is zero.
    - **Small**: If the word count is between 1 and 50 (inclusive).
    - **Big**: If the file contains more than 50 words.

Write a shell script that performs the following tasks:

1. Prompt the user for a directory name and assign it to a variable `directory_name`.
    - If the directory exists, proceed to the next step.
    - If not, print `'No such directory'` and exit.
    - *Hint*: You can use `-d $directory_name` inside an `if` condition.

2. Change into the given directory. Prompt the user for a **file extension** (e.g., `txt`, `log`, `sh`) and store it into a variable called `file_ext`.

3. Use a `for` loop to iterate through all files in the directory and its subdirectories that match the given extension and are writable by the current user.
    - For each such file, print the following: i) File name, ii) Character count, iii) Size class

4. The size class is defined as follows:
    - **Empty**: If the character count is zero.
    - **Small**: If the character count is between 1 and 250 (inclusive).
    - **Big**: If the file contains more than 250 characters.