```
gcc program.c # Compiles program.c and generates a default executable named a.out
./a.out # Runs the compiled program
```

```
gcc program.c -o my_program  # Compiles program.c and creates an executable named my_program
./my_program # Runs the compiled program
```

```
gcc program.c -o input_program # it takes inputs from input1.txt and displays the result in the terminal.
./input_program < input.txt
```

```
gcc program.c -o io_program
./io_program < input.txt > output.txt
# it takes the inputs from input1.txt and creates the myoutput.txt document and writes the result here.
```

```
diff output1.txt myoutput.txt # compares the document output1.txt with the document myoutput.txt.
```

```
diff --ignore-all-space output1.txt myoutput.txt
# compares the document output1.txt with the document myoutput.txt without ignoring the spaces.
```

QUESTION 1

Write a program that groups the characters in a given string alphabetically and prints how many times each character appears. The order of characters doesn't matter, but the output should be in alphabetical order.

1. Function Implementation: The program defines a function named countCharOccurrences() that counts the frequency of each character in the given string. This function is called from the main() function.

2. Input: A string is provided by the user.

3. Output: The frequency of each character is printed in alphabetical order.

4. Constraints: The order of characters does not matter, but the output should be in alphabetical order.

Input1: Enter a string: banana

Output1: a3b1n2 **//(The character 'a' appears 3 times.**

The character 'b' appears 1 time.

The character 'n' appears 2 times.)

Input2: Enter a string: appepie

Output2: a1e1i1p3