Test Cases

- For the validateString class:
  - valid string:
    - This test method checks whether the validateString function returns true when passed a valid string ("NAME"). It should assert true because "NAME" is a valid string.
  - invalid string:
    - This test method checks whether the validateString function returns false when passed an empty string (""). It should assert false since an empty string is invalid.
  - a jump line:
    - This test method checks whether the validateString function returns false when passed a string containing a newline ("\n"). It should assert false because the presence of a line break makes the string invalid.
  - a single character:
    - This test method checks whether the validateString function returns false when passed a single character string ("O"). It should assert false since a single character string is invalid according to your criteria.
  - Invalid minimum size:
    - This test method checks whether the validateString function returns false when passed a string ("LONGCHAR") and a minimum required size (100). It should assert false because the string does not meet the specified minimum size.
  - oneJumpLineValid:
    - This test method checks whether the validateString function returns true when passed a string ("LONGCHAR") and a minimum required size (4). It should assert true because the string meets the specified minimum size.
  - singleCharMinimun:
    - This test method checks whether the validateString function returns true when passed a string of a single character ("O") and a minimum required size (1). It should assert true since the string meets the specified minimum size.
  - emptyWithMinimum:

  - This test method checks whether the validateString function returns false when passed an empty string ("") and a minimum required size (4). It should assert false because the string does not meet the specified minimum size.

- For the posCodeTest class:

  - Valid zip code capitalization:

  - Tests whether the validPostalCode function correctly converts to uppercase and returns a valid uppercase postal code ("M2K 1H7").

  - valid zip code without capital letters:

  - Tests whether the validPostalCode function converts to uppercase and correctly returns a valid lowercase postal code ("m2k 1h7").

  - Valid Postal CodeMixedCaps:

  - Tests whether the validPostalCode function capitalizes and correctly returns a valid mixed-case postal code ("m2K 1h7").

  - valid zip code without space:

  - Tests whether the validPostalCode function adds missing spaces and correctly returns a valid postal code without spaces ("M2K1H7").

  - validpostalcodemultiplespaces:

  - Tests whether the validPostalCode function removes extra spaces and correctly returns a valid postal code with multiple spaces ("M 2K 1 H 7").

  - invalid zip code:

  - Tests whether the validPostalCode function returns NULL when passed an invalid postal code (" 1H7").

  - special characters:

  - Tests whether the validPostalCode function returns NULL when passed a postal code with special characters ("!2%1@7").

  - Empty:

  - Tests whether the validPostalCode function returns NULL when passed an empty string ("").

Reflextion

1. To ensure that no blank lines were entered when collecting customer information, I implemented a validation process during data entry. I iterated through each character in the input and checked if it was whitespace. If it found a blank space, it ignored it and continued collecting the information. It only stored non-whitespace characters in a new string.

   However, if you needed to test that request messages are generated correctly in case of blank lines, you could implement additional unit tests. You could use testing frameworks like Google Test or Catch to redirect the stdout to a temporary file. Then you would run the program and compile the output into that file. It can then analyze the file to verify that the prompt messages are generated correctly when a blank line is entered. This would allow me to confirm that the program produces the correct messages in response to blank line input.