Algorithm for the shorthand notation program
Siqi Wen 2021 CS162

Step 1 - Welcome the user

1a. This step will be created in a function, which will be called in main;

1b. Output the welcome message: “Welcome to the shorthand notation program.”;

1c. Introduce the main purpose of the program by outputting: “You'll need to set up three words with corresponding abbreviations, and you can enter in your notes in shorthand notation, then the program will replaces those abbreviations with actual words in the sentence.”;

Step 2 - Set up the three words and abbreviations

2a. Create six functions for setting up the three words and abbreviations;

2b. Ask for the three words and abbreviations by prompting the user: “Please enter the first word: ”; “Please enter the second word: ”; “Please enter the third word: ” and so on;

2c. In each function, read in the answer, and store them in an array variable;

2d. At the end of the function, return the array (address);

2e. In the main, echo the answers out, and ask the user if they want to re-enter the words by prompting: “These are what you’ve set up: xxx(first abbreviation) stands for xxx(first word); xxx(second abbreviation) stands for xxx(second word); xxx(third abbreviation) stands for xxx(third word). Do you want to re-enter?(Y/y or N/n): ”;

2e. i. This “re-enter” process will be created in another function and the function would return “true” or “false”;

2e. ii. If the user entered something that is not “Y/y” or “N/n”, then the program would keep prompting them until they enter “Y/y” or “N/n”;

2e. iii. If the user entered “Y/y”, then the function would return “true”, otherwise, “false” will be returned;

2f. If the function returned “true”, then the program would repeat Step2, otherwise, continue to next step;

Step 3 - Prompt for the sentence

3a. Ask the user to enter in a sentence representing their notes in shorthand notation, store the sentence in an array variable;

3b. This will also be a separated function, the sentence the user typed will be the argument that will be passed into the function;

Step 4 - Analyze the sentence and output it with actual words

4a. Use strlen() function to get the length of the sentence, and store it in a variable before the for loop;

4b. Use a for loop to step through the array of the characters for the sentence;

4c. Look for the parts which match with the abbreviations, and replace the abbreviations with the actual words; if the abbreviation is capitalized, then the actual word should be capitalized; otherwise, it should stay lowercase;

4d. Output the sentence with actual words instead of the abbreviations;

Step 5 - Ask if the user wants to do it again

5a. Ask the user if they would like to do this again;

5b. Use the same re-enter function appeared above;

5c. If the function returned “true”, then the program would repeat Step2, otherwise, continue to next step;

Step 6 - Output the ending message and terminate the program

6a. Output the message: “Thank you for using our program! Have a wonderful day!”.