Zakeriya Muhumed
Algorithm Program #2
CS162 PSU

Algorithm for Password Generator

1. Welcome the user to your program
    a. Introduce the basic instruction and the use of your program.
        1. Prompt "Welcome to Password Generator"
        2. Message with everything about the program.
            a. "THis is a password generator program, you will enter in a word and we will modify it for you
    b. Notify the user the different options and basic controls of the program.
        i. Prompt "Once the program is done running you will have an option to quit or it will rerun".
    c. Have a specific input in the program for different user interfaces.
        i. Prompt message about the 2 user interfaces.
        ii. Prompt (A, B as user interfaces choices).
            a. Option one being we only show your final password.
                i. Prompt a message explaining this.
            b. Option two is modify the password to your liking after the password generator.
                i. Prompt a message explaining this.
    d. After the user has picked the different user interfaces, proceed to step 2.
2. Obtaining the user information.
    a. Inside the input function
        1. Prompt users to enter their first name and last name.
            a. Use an input function to get the user input.
            b. Store this information inside the struct name array.
        2. Prompt the user to enter their email address.
            a. Use an input function to get the user input.
            b. Stoer this information inside the struct of the email array.
        3. Prompt users to enter their password
            a. Prompt "Enter in a password"
            b. Uses an input function to get the user input.

i. Store this information in the password array and use cin width to correctly store.

4. Error check
    a. Use the cin dot width and ignore
    b. If the input type is not right, reenter.
    c. Echo back their inputs.

3. Modify
    a. Inside mod function
        1. First step of modification is to check the capital letter if it is capital. Do the following.
            a. Use a for loop to check the capitalization of a letter once it is found.
            b. Place an underscore (__)before every capitalization letter.
            c. Follow that up by lowering the case version of that letter.
            d. Use the cctype library for this.
            e. Expand the password generator for the next modification.
        2. Check for vowel, if you found them inside the array, do the following
            a. Duplicate that vowel right next to each other.
            b. Get this to work before implementing the next modification.

4. Change
    a. Inside the change function
        1. Changing the last character
            a. Take in the last character and convert it to ASCII values.
            b. Use the int of the the last character to find the ASCII value.
        2. If there is no capitalization of any character.
            a. Add a symbol of my choice to the end of the password.
            b. Use the symbol$ at the end of the password.

5. Unique change, everything will be under a struct
    a. Inside the struct and other function
        i. The user name, email and their password.
    b. Compare the password array with the name array and email address array
        i. If any similarity are found
            a. Ask the user to re-enter
                i. Prompt this message, “Password can’t contain name or email”
            b. Use this in a loop until the user has entered the correct password.

6. Display
    a. End of main function
        i. Display all the information depending on the user interface option.
7. ENDING
    a. Beginning of main function
        i. Start with a do while loop.
        ii. This do-while loop will execute everything in main at once.
        iii. Users will get a prompt with the option whether they want to quit or not.
        iv. Prompt “Enter Q to quit, else the program will rerun!!”.
        v. Until the user wants to quit the program will keep rerunning.