**Name:**

**Programming Challenges for Chapter 5 – File Access**

This selection of challenges exercises our knowledge of loops. You may use whatever loop you wish but keep in mind the discussion from Chapter 5, Section 5.9, which summarizes each loop type and the situations in which the loop works best.

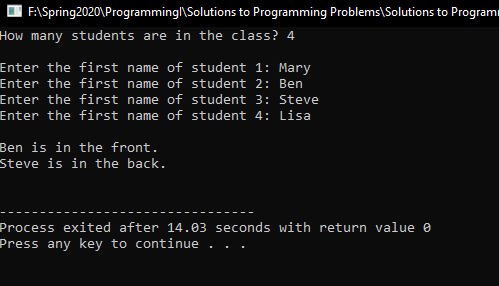
We will also practice using file access. The files you need for this assignment are included on Blackboard.

**Challenge 1:**

*Student Line Up*

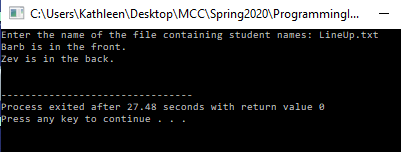
A teacher has asked all her students to line up alphabetically according to their first name. For example, in one class Amy will be at the front of the line, and Yolanda will be at the end. Write a program that prompts the user to enter the number of students in the class, then loops to read the input from the user for that many names. Once all the names have been read, your program reports which student would be at the front of the line and which one would be at the end of the line. You may assume that no two students have the same name. **Input Validation**: Do not accept a number less than 1 or greater than 25 for the number of students.

The output will look something like this:



**Challenge 2:**

Modify the **Student Line Up** program described in Programming Challenge 1 so the program retrieves the names from a file. Have the program ask the user to input the filename into a variable and use that variable to open the file. Check to make sure the file opens successfully. Names should be read in until there is no more data to read. The file named **LineUp.txt** is included in the Chapter 05 Lab folder for this exercise. You can use this file to test the program. **Note:** Read the Chapter 5 sections on *Letting the User Specify a Filename* and *Using the c\_str Member Function in Older* *Versions of C++* carefully because some of you will need to use c\_str function to be able to open the file.



**Challenge 3:**

*Numeric Processing*

You will find a file named **Random.txt** in the Chapter 05 Lab folder. This file contains a long list of random numbers. Copy the file to your system, then write a program that opens the file, reads all the numbers from the file, and calculates the following:

* The number of numbers in the file
* The sum of all the numbers in the file (a running total)
* The average of all the numbers in the file Budget Analysis

Your average should display no more than 3 decimal places. The output will look something like this:

