C#.NET LAB 9: REFACTORING TIME!

NOTE: Points will be awarded for items that are written correctly in themselves but don't actually work because other things are broken. There is a total of 10 points available for this lab.

Task: Improve your student information system from the previous lab by using a collection.

Build Specifications:

- **1 Point:** Refactor your code to use Lists rather than arrays.
- **2 Point:** Add another list to store another piece of information (so perhaps favorite color or favorite number) about each student. Update your validation to allow the user to select the third piece of information.
- **1 Point:** Each iteration of the loop, ask first if the user would like to find out information about a student or add another student.
    - **1 Point:** If they choose to add another student, get the name and each piece of information and add them to the end of the list.
    - **1 Point:** Validate input -- do not accept blank answers for any of the questions.

Additional Requirements:

- **1 Point:** For answering Lab Summary when submitting to the LMS
- **-2 Points:** if there are any syntax errors or if the program does not run (for example, in a Main method).

Extended Exercises:

- **1 Point:** Create the original lists in alphabetical order by student name. When a user adds a new student, insert them at the correct location alphabetically. Remember to put the information about a particular student at the same index in each list!
- **1 Point:** If you already know about creating classes, go back and rewrite this:
    - Make a StudentInfo class with name and other information (hometown, food, whatever you chose) as data members.
    - Use a single List of StudentInfo instances to store the information.

E: 10-9 **M:** 7 - 8 **D:** 0 - 6