Group Project Checklist

Programming / Functionality

- Declare an array of doubles
- This array must have the following values (in order): (137.50, 138.25, 130.50, 150, 150, 162.50)
- You must have a loop in main() that prompts the user for each of the people
  - the person's name
    - if the name is empty (i.e. ""), display an error message and skip this person
  - the day in which they are checking in to the hotel
    - all days must be implemented as C-Style strings (char arrays)
  - the day in which they are checking out of the hotel
- All prompts for user input should not move the cursor to a new line.
- For each person, display their name and their total room cost.
- At the end, display the total costs for the entire group.
- You must create a function called indexTheDay(). It converts from the user input to a useful array index. It is called from main().
- The -1 error value must be represented by a global constant. It must go before any function definitions in your code. The constant's name must adhere to the SET Coding Standards.
- create a function called calculateCostOfRoom()
  - It must take exactly these parameters, in order:
    - the array of room rates
    - the day of check-in, as an index into the array
    - the day of check-out, as an index into the array
  - it must return the total cost for the room, as a double.
  - It must do no output.
  - There is no need for error-checking in this function.

Things that help me

- The only blank lines should be present after each person is completed. Normal output should fit within the normal Visual Studio console window.
- Use fgets() for input.

- There are five errors that you need to handle in this program:
  - Missing name
  - Invalid check-in day
  - Invalid check-out day
  - Invalid length of stay
  - Missing the meeting

Style

- make sure indentation is correct

Commenting

- ~~header comment~~
- ~~create checklist.pdf (however I want)~~

File naming

- project is called major3
- source is called m3.cpp

Other things that are required in all assignments

- return 0 in main()
- braces around single-line bodies for if and while and for, etc.
- initialize all variables when declared

Submitting

- submitting:
  - clean solution