Part 5

A. (2 pts) As you worked to implement your Treasure Chest, you probably thought of new test cases after your Design Document was already submitted. List one here. (If you didn't think of any new test cases while implementing, create one now.)

Entering an empty name, since it is getline, it should accept it. Accepting it defeats the purpose of entering a name therefore it should let the user enter again.

B. (9 pts) Try each of your 9 test cases (there are 9 now) and for each one, report whether or not the behavior is as expected. If not, state whether (1) your design has evolved and now the expected behavior is different (state what the new expected behavior is) or (2) this test helped you find (and fix) a bug in your program.
   - If you didn't have 8 test cases in your Design Document, make them up now to allow you to get full credit here.

It helped me to find a conditional error within my function that validates for the index of user input with the size of the array which I included one index beyond the true to be valid and it caused seg fault. I fixed it by making it > instead of >=.

C. (4 pts) Introduce a bug in your program that causes it to crash when it runs (e.g., invalid access into your 2D array). Compile your program with -g and then run your program in gdb (see Lab 8). When the program crashes, copy and paste the error that is generated into your document. Then generate a stack trace and copy and paste this stack trace into your document.
   - Does gdb's output lead you to the right program line where you introduced the bug?
   - Write down the order of function calls that led to the place where the program crashed. (e.g., func2 -> func2 -> func3 …)

(gdb) bt

#0 std::string::empty (this=0x71)

at /usr/src/debug/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/x86_64-redhat-linux/libstdc++-v3/include/bits/basic_string.h:812

#1 0x0000000000401f5a in main () at assign5_treasure.cpp:181

We can see that at line 181 there is an error and when breaking there we can see it allows an access at [3][1] when the array size is 3 x 3.

This is in the main and no functions are called yet.

Part 6

- (up to 2 pts) Add another menu option (beyond the 5 that are required) that lets the user do something else with their collection. Document this in your README.txt file.

DONE

- (up to 2 pts) Add a luck aspect to the game. Each time the user makes a menu choice, generate a random number to see if a luck event (something outside the user's control) happens (you choose how likely this is to happen). Your luck event could be good or bad. Examples (these are ideas; you only have to implement one, not all):
    - One of their items breaks (is destroyed). If so, notify the user and update the treasure chest to remove that item.
    - One of their items doubles in value. If so, notify the user and update the item's value member.
    - Your idea here!

DONE

- (up to 2 pts) Ask another person (friend, family member, roommate, coffee shop stranger) to try out your program. In “Part 6” of your Word/text file (created in Part 5), copy/paste their full interaction (program outputs and what they typed). Did the program crash or behave in a way you did not expect? Describe anything you found surprising.

Welcome to the Hotel Room Treasure Chest!

Enter number of rows: 5

Enter number of columns: 6

5 |_|_|_|_|_|_|

4 |_|_|_|_|_|_|

3 |_|_|_|_|_|_|

2 |_|_|_|_|_|_|

1 |_|_|_|_|_|_|

Total value: 0

1) Add room key

2) Remove key

3) Add random keys

4) Show key

5) Economic shift

6) Edit

7) Quit

Please make your choice: 1

Enter row: 1

Enter column: 1

Enter a name for your room: Blue

Enter which floor the room Blue exists on: 34

Enter the number of beds in your room: 69

Enter the cost of your room: 4567

Does your room have a tv?(0-no, 1-yes) 1

5 |_|_|_|_|_|_|

4 |_|_|_|_|_|_|

3 |_|_|_|_|_|_|

2 |_|_|_|_|_|_|

1 |B|_|_|_|_|_|

Total value: 4567

1) Add room key

2) Remove key

3) Add random keys

4) Show key

5) Economic shift

6) Edit

7) Quit

Please make your choice: 3

Enter row: 4

Enter column: 2

Enter a name for the room you would like to generate: Gold

5 |_|_|_|_|_|_|

4 |_|G|_|_|_|_|

3 |_|_|_|_|_|_|

2 |_|_|_|_|_|_|

1 |B|_|_|_|_|_|

Total value: 4866.01

1) Add room key

2) Remove key

3) Add random keys

4) Show key

5) Economic shift

6) Edit

7) Quit

Please make your choice: 4

Enter row: 1

Enter column: 1

Name: Blue

Floor: 34

Beds: 69 per night

Cost: 4567

TV: included

5 |_|_|_|_|_|_|

4 |_|G|_|_|_|_|

3 |_|_|_|_|_|_|

2 |_|_|_|_|_|_|

1 |B|_|_|_|_|_|

Total value: 4866.01

1) Add room key

2) Remove key

3) Add random keys

4) Show key

5) Economic shift

6) Edit

7) Quit

Please make your choice: 5

The economic shift made prices to be 3.71% of their original price.

5 |_|_|_|_|_|_|

4 |_|G|_|_|_|_|

3 |_|_|_|_|_|_|

2 |_|_|_|_|_|_|

1 |B|_|_|_|_|_|

Total value: 5046.54

1) Add room key

2) Remove key

3) Add random keys

4) Show key

5) Economic shift

6) Edit

7) Quit

Please make your choice: 6

Enter row: 1

Enter column: 1

Enter a new name for your room: Green

5 |_|_|_|_|_|_|

4 |_|G|_|_|_|_|

3 |_|_|_|_|_|_|

2 |_|_|_|_|_|_|

1 |G|_|_|_|_|_|

Total value: 5046.54

1) Add room key

2) Remove key

3) Add random keys

4) Show key

5) Economic shift

6) Edit

7) Quit

Please make your choice: 7

Everything behaved as normal as I coded it. It did not crash, but my friend did not try to insert any invalid inputs and that might be the reason why, but personally I've tried to enter invalid inputs and boundary test cases and they all were handled unless if a letter or symbol is inserted into an int input.

- (up to 2 pts) Describe (in "Part 6" of your Word/text file) one improvement you would recommend to make your program better (more robust, more entertaining, more attractive, anything).

Allows the user to copy or move items or have menu take in char so user can type in any char and will be prompted invalid instead of it crashing. I could also implemented an edit function.