

Homework 3

flow of control: if, if-else, switch, while, command line arguments

July 10, 2019

Abstract

The objective is to give you practice with the statements by which we control the flow of execution, and allow data to modify the flow of execution. We will also navigate through a graph data structure. User input will be collected and used.

The context of this problem is carrying out a hunt for treasure in a “house” that is represented by a graph. The program will accumulate treasure, and keep track of how much is found in what place. The program will accept user input to customize the search process.

Be sure to construct test cases for each function you build. Put thought into the test cases. The goal is to know that a function works, when it passes all of its test cases. One test case per function is usually not enough.

- Construct a sequence diagram for your project. You can draw it by hand and include a photo, or draw it with a software tool.
- Use the test-driven development style for developing your code. Document this by taking a new screen shot of your test code every time you make a new test (See Figure 1.), and of your production code every time you extend your production code. It could be a large number of screen shots. Be sure to run your code every time you add a few lines of test or production code. (See Figures 2 and 3.) Do not allow the number of errors to get large.
- An explanation of a representation of a graph will be given here, because a graph data structure will be used to represent the layout of the house. The program must use this data structure to present to the user the choices of next rooms to search. The representation is called an adjacency matrix. It is a two dimensional matrix. The number of rows and the number of columns are both set to be the number of nodes in the graph. In our case, it is the number of rooms in the treasure house. The values in the matrix come from whether the room number that is the row number is directly connected to the room number that is the column number. Because these direct connections are bidirectional, the adjacency matrix is symmetric about its main diagonal. Moreover, the values in the main diagonal are

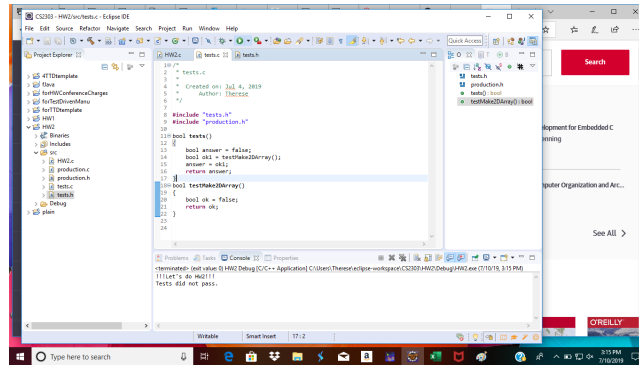


Figure 1: In the process of creating a new test.

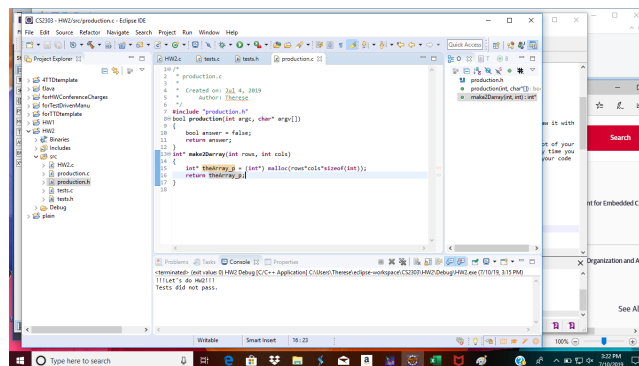


Figure 2: Starting some production code.

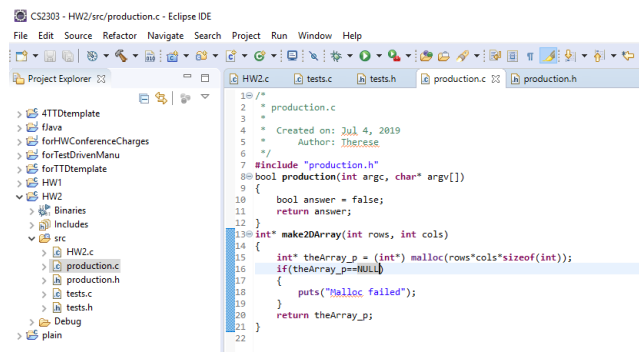


Figure 3: Adding more production code.

2. Create the sequence diagram and include the electronic file (diagram, screenshot or photo). Make sure your name appears within the sequence diagram.
3. Place function prototypes for all of your functions from the sequence diagram into one or more .h files.
4. As you work on the assignment, collect a sequence of screen shots showing how your test code, and your production code are growing.
5. Be sure to build and run often; do not allow errors to build up.
6. Show the sequence of moves from room to room by listing them, and also show the subtotal of treasure through the house.
7. Receive the values entered on the command line. Ask for values not provided in the command line. In either case, check for reasonable values. Negative treasure or rooms implies no search should be conducted. Use these limit values in execution.

Grading

Criteria	Possible Points
Project that looks like starter code	20
Sequence diagram that reflects the problem statement	20
Documentation of code development (those screenshots) that clearly follows test-driven style	20
Screenshot of rooms searched, visited and treasure collected subtotals, that correspond	20
Correct treatment of command line arguments	20
Total	100