Section 1:

1. Why it is good to write test cases
   a. Outlines what the program is meant to do and the output that is meant to occur with a given input, makes sure you have addressed all the requirements
   b. It allows for future testers to run test cases on program
   c. It reduces bugs, provides quality assurance, makes both user and programmer more confident in the program's reliability
2. Use of mocks in testing
   a. Mock testing is generally used in unit tests and is a way to replace dependency in these tests

- Advantages:
    - Useful when needing to isolate specific components: Being able to test a single method on a single object that interacts with other objects, without having the dependencies affecting the function of the method tested
    - Localise testing: allows you to run tests using fake database connections, without having to commit to particular input databases, lightweight emulation of the system to give initial definition of required functionality
- Disadvantages:
    - You need an understanding of the dependencies of the methods objects you are testing, otherwise the mock won't accurately represent behaviour
    - It can increase the amount of maintenance needed to perform the tests themselves, it isn't flexible and mocking can demonstrate that the design has too many dependencies

3. Where insufficient code testing led to problems
   a. Therac- 25, a radiation therapy machine which offered 2 treatments, (1) firing beam of low energy electrons or (2) delivering radiation. The separation of these two modes were software controlled.
   b. Due to lack of unit testing, a malfunction occurred from a bug where both data entry and keyboard handler routines shared a single variable.
   c. Between 1986 and 1987 the machine killed 4 patients and left 2 injured (Fabio, 2015) since machine in wrong mode and deliver radiation

Fabio, A 2015, *Killed by a Machine: The Therac-25,* viewed 25 May 2020, https://hackaday.com/2015/10/26/killed-by-a-machine-the-therac-25/

Section 2:

1. BFS
   a. gives the user the shortest route to the end (most efficient), this is useful to compute the smallest number of moves needed to solve a puzzle.
   b. It consumes more memory as connected nodes need to be stored in memory and can be slower
2. DFS
   a. gives the user a route the fastest compared to BFS (when the ending cell is large distance from start) but isn't necessarily the most optimal solution
   b. It is useful it when path is short and when needing to find a route.
   c. Consumes less memory but may get trapped in searching a useless path.
3. If ending cell is very close to starting cell, BFS is faster otherwise DFS is faster.
4. The game set-up has cells such as teleport, water and fire which means you may have to revisit certain cells in order to reach the end, meaning that list of cells visited won't work. Eg. Need to use teleport to get water bucket and the use the teleport again (revisit the cell) to reach the end point.