Project 1 FAQ:

Is it possible for our King to capture an enemy piece? **Answer: No. Our King cannot capture any pieces as stated in the PDF.**
Do we have to only return the moves from the path that leads from our source to our goal, or all the moves we do in the process of getting there? **Answer: We only return the list of moves that is the path from the start position to the goal position**
Do we implement Graph or Tree Based BFS/DFS/UCS/A*? **Answer: It is up to you to decide based on the tradeoffs of tree search and graph search.**
How fast in general should my algorithm be? **Answer: In general, you should be able to take < 1s for all testcases, with some testcases taking <0.1s.**
Will there be a testcase where the input itself is inconsistent and do we have to handle errors? E.g. it says 20 obstacles but gives 21 positions. **Answer: No. All the testcases given are valid scenarios. You only need to account for situations where a path to the goal may not exist (etc. having obstacles completely blocking the goal). In this case, you will have to return an empty list as stated in the pdf.**
Can we assume step cost is at least 1? **Answer: Yes, if the testcase file does not state the action cost for the particular position, you can set the action cost for the position as 1. Additionally, all step costs are in integers.**
Does “ugly code” result in mark deduction? **Answer: Generally, we won't deduct marks for any "ugly" code as we emphasis more on the correctness of your algorithms and the implementations of the code (This is not a Software Engineering mod). The random check will mainly check for your logic and technical correctness.**
How will our marks be given? **Answer: Generally, if you pass all the testcases on CodePost, you should get full credit for that particular search algorithm. Manual inspection will mainly check if you are not using any bruteforce method and checking the logic of your code. If your logic is not done correctly (etc using DFS in your BFS code), marks will be penalized even if you pass all testcases.** **You are not graded on writing clean code/having good practices although this is highly encouraged for the readability of it.**
Can we add some Python files to contain common data structures (without additional subfolder)? **Answer: No. Only submit the required files as stated in the PDF. This is to ensure consistency in submissions.**

Is it safe to assume that there will only be 1 King (i.e. 1 starting position) and no other friendly pieces for all inputs? **Answer: Yes.**
I tried to run the test cases on my AStar implementation for about 10 times, and it passed all test cases for about 7 times but occasionally failed on test case 1 or 3, can I ask what can I do to deal with this? **Answer: If you passed all testcases in a submission on CodePost, you should be fine and leave your submission as it is. The time limit on CodePost has a variance which we cannot control due to their internal server.**
How should my program handle the case when the goal position is also occupied by another obstacle - or that the goal position is not occupied by anything but is in a threatening position of an enemy piece? **Answer: If there is such a case where the goal cell is threatened or occupied, your King should avoid that goal cell. It may or may not return empty depending on whether there are any other valid goal cell(s).**
Can I use libraries that are not stated in the pdf? **Answer: No.**