CPSC 2150 – Algorithms and Data Structures II

Programming Project: Chess Knight

Total - 40 Marks

Learning Outcomes

- Problem solving with trees
- Develop efficient C++ code to solve a problem.

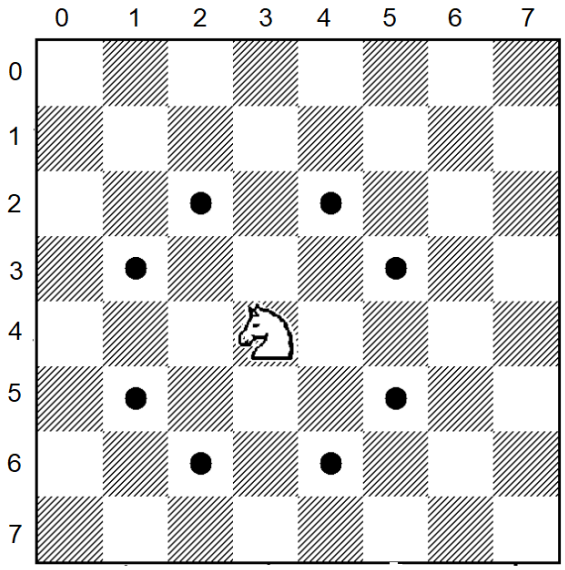
Resources

- Chapter 7 of the text book

Description

A knight in chess can move to any square on the standard 8x8 chess board from any other square on the board, given enough turns (don't believe it? See this animation). Its basic move is two steps forward and one step to the side. It can face any direction.

All the possible places you can end up after one move look like this:



Your task is to build a function knight_moves that shows the simplest possible way (path) to get from one square to another by outputting all squares the knight will stop on along the way. By "simplest way", I mean the path with the least move.

You can think of the board as having 2-dimensional coordinates. Your function would therefore look like:

- knight_moves([0,0],[1,2]) == [[0,0],[1,2]]
- knight_moves([0,0],[3,3]) == [[0,0],[1,2],[3,3]]
- knight_moves([3,3],[0,0]) == [[3,3],[1,2],[0,0]]

Your program must ask for the dimension of two valid squares in the chess board (8x8) and then prints the total number of moves with the details such as:

```
=> Enter the current Knight’s location: 3 3
=> Enter the destination location: 4 3
=> You made it in 3 moves from [3,3] to [4,3]!  Here is your path:
    [3,3]
    [4,5]
    [2,4]
    [4,3]
```

SUBMIT to D2L

Submit a zip file named **StudentNumber-Project-ChessKnight.zip** including all related files by the due date. For example, if your student number is 10023449, the submitted file must be named as **10023449- Project- ChessKnight.zip**.