**Loh Hao Bin (25461257) & Tan Wen Jie (25839063)**

**Test Cases (Task 1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Description** | **Test Steps** | **Expected Results** | **Results** |
| 1 | Test the menu | 1. Run 2. Enter 1 | The program should ask for x and y input | Pass |
| 2 | Test the menu | 1. Run 2. Enter 2 | The program should quit while displaying “User stopped the program” | Pass |
| 3 | Test for valid input | 1. Select 1 2. Enter 2 for both X and Y input | The current position of the knight is mark as “K” while the previous position of the knight is mark as “\*” | Pass |
| 4 | Test for invalid input for position | 1. Select 1 2. Enter 9 for both X and Y input (which are out of bound) | The current position should remains the same, while returning “Please input a valid File and Rank.” | Pass |

**Test Case (Task 2)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Description** | **Test Steps** | **Expected Results** | **Results** |
| 1 | Test the start menu | 1. Run 2. Enter 2 3. Input 7 for both x and y | 1.The knight should initially be a position (1,0).  2. After the start position is inputted, the knight should be at (7,7) on a new board. | Pass |
| 2 | Test the next possible position | 1. Run 2. Enter 3 | While “K” is positioned at (1, 0), a list of possible legal next moves should be printed. [1, 3], [2, 2], [2, 0], note that illegal positions are not displayed. | Pass |

**Test Case (Task 3)**

**For the Test Cases in Task 3, we are using the file knight\_tour\_2.py for the tests.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Description** | **Test Steps** | **Expected Results** | **Results** |
| 1 | Test for exception in start menu | 1. Run 2. Enter 9 | 1.The knight should be a position (1,0).  2. Message ”Please input a valid command”  3. The board is printed again | Pass |
| 2 | Test for exception in start menu | 1. Run 2. Enter 2 3. Input 9 for both x and y | “K” remains the same with message “Please input a valid File and Rank within range.” | Pass |
| 2 | Test for exception in position menu | 1. Run 2. Enter 3 3. Input a number that is not within the list (e.g 9) | “K” remains the same with message “Please choose from the valid position above.” | Pass |

**Test Case (Advanced)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Description** | **Test Steps** | **Expected Results** | **Results** |
| 1 | Test for undo functionality | 1. Enter 3, and then input another position to move to [3,3]  2. Enter 5 to undo | 1.The knight should move to [3,3]  2. After undo it should return to the previous position, and its ‘next position’ shouldn’t have a \* | Pass |
| 2 | Test for undo when there’s nothing to undo | 1. Start the program, and enter 5 to undo | It should output “There’s nothing to undo” | Pass |
| 3 | Test for Solution | 1. Enter 4 | It should return a traversed board with an open knight’s tour, and return the sequences. | Pass |
| 4 | Test for solution when there’s no further solution | 1. Move the knight further until into a position where it is certain to have a deadlock  2. Enter 4 | It should return “Error: There's no solution from this position onwards.” | Pass |