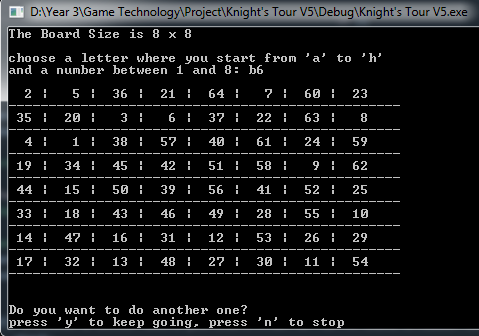
Knight's Tour

**Dylan Gijsbertsen**

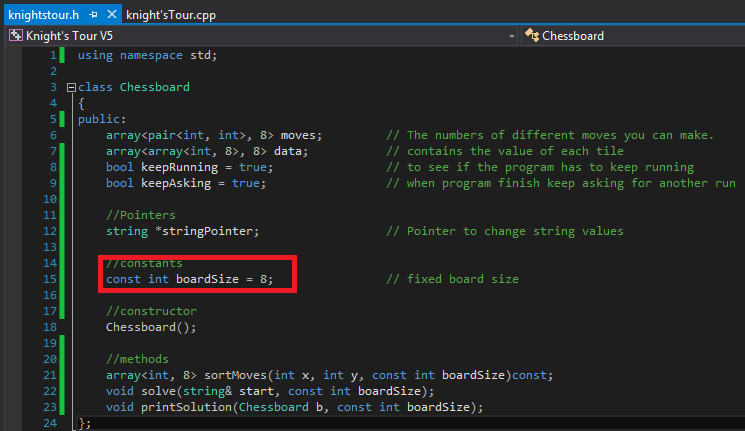
**500687199**

**Requirements:**

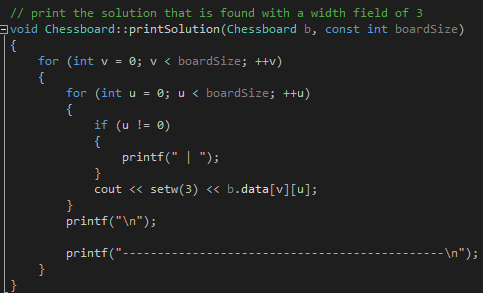
1. **Programming language has to be C++**
2. **The program makes use of pointers and references**
3. **There is a visual representation**



1. **the board is at least 5 x 5 in size**



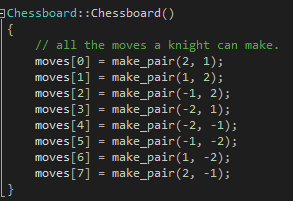
D:\Year 3\Screen shots\boardSize in method call.png



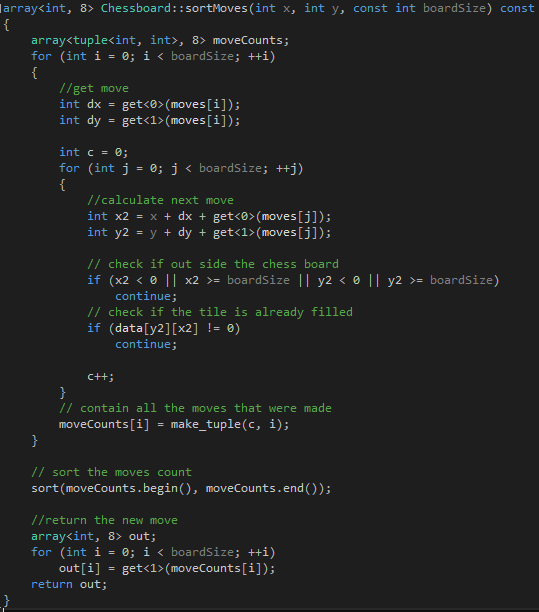
1. **The knight has to move according to its movement rules**

D:\Year 3\Screen shots\moves array.png

This array moves has a length of 8 because a knight can make 8 different moves. For every index place I make a pair of two integers for a combination that the knight can move.



In the constructor I set for every index a move that a knight can make by pairing them together.

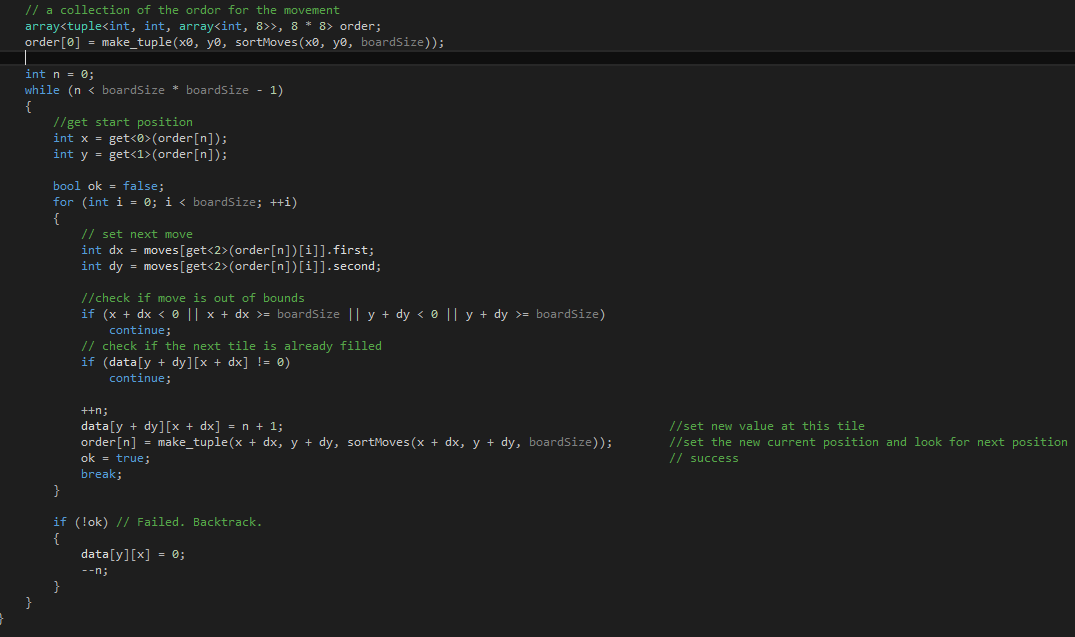


The moveCounts keeps track to all the moves that the program made. First I calculate the next move for the next position, and that calculation is the current x and y position + the move dx and dy + another move to see the next position.

Then I check if the calculated move is out of the board or the tile is already filled.

Than all the successfull moves are stored in the moveCounts variable and sort it.

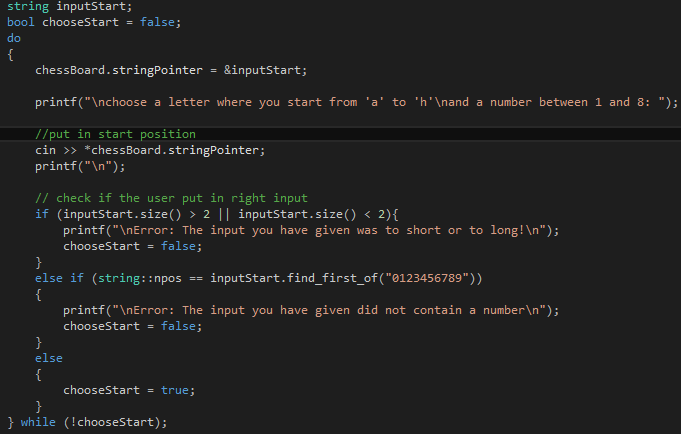
Now I return



1. **The knight can only visit each square once**

**Exemplar:**

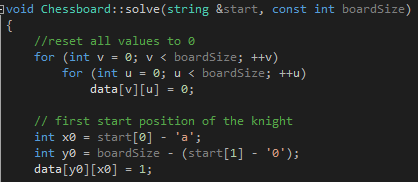
1. **Board size can be set by the user**
2. **The user can decide where the knight starts**

****

**First I tell the user to give a letter between ´a´ and ´h´ and also set a number between 1 and 8. Just like on a chessboard you have than a1,b1,c1 etc. After that I check if the input you gave is the correct input to start the solution. as long you put in the wrong input it will keep looping through it untill you get it right.**

**D:\Year 3\Screen shots\boardSize in method call.png**

**Now that you have put in the correct position of the grid. In the solve method I send a argument with the input string and the board size to solve the tour.**

****

**The calculation for x0 is been done by getting the first index from the string start and subtract the charcter 'a'.**

**example:**

* **string input = a1;**
* **x0 = 'a' - 'a' = 0**

**So x0 = 0, and that is the index number for *data* to set the x position.**

**To calculate y0 you get the second index from the string start and subtract '0' and than with that result you subtract that from boardSize.**

**example:**

* **boardSize = 8;**
* **string input = a1;**
* **y0 = 8 - (1-'0') = 7;**

**So y0 = 7, and that is the index number for *data* to set the y position.**

**Now that we have the calculated the x0 and y0 we set that position in *data* and give that the value of 1. This will result that the starting position with the input ´a1´ will be at the left down conner.**

1. **Uses a more efficient solution than brute force**