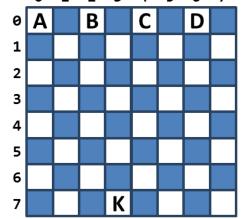


High-Quality Code Construction – Game "King Survival"

Your task is to write an interactive **console-based implementation of the game "King Survival"** in which **the King (K) tries to escape the four pawns (A, B, C, D)**. The game is played on a standard **chess-board of size 8 x 8 gells**(half of the mylite the other half black).



The King initially is located at **row 7 and column 3** and the pawns are located at row 0 and columns 0, 2, 4 and 6. A pawn can only move at neighbor cell **down-left** or **down-right**, while the King can go to neighbor cells in **4 directions**: down-left, down-right, up-left and upright. The pawns and the King cannot step outside the chess-board or on a cell that is already occupied. The game is first started by moving the King. At the odd turns the player **moves the King** and at the even steps the player **moves the pawns**. Thus the player in fact plays against himself. To make a move the player issues a command consisting of the name of the figure, i.e. "**K**", followed by the direction to move, e.g. "

KUL" means King-Up-Left, "ADL" means Pawn-A-Down-Left. Thus all possible commands are 12: KUL, KUR, KDL, KDR, ADL, ADR, BDL, BDR, CDL, CDR, DDL and DDR.

Write a program that **simulates the "King Survival" game** as shown in the sample game session. Initially the game starts with the initial board (shown at the figure above) and it's King's turn. After the King moves, its Pawns' move and so on. Before each turn the board is visualized as shown below.

If the King reaches row 0 **he wins**. If the pawns "catch" the King so that he has no valid moves at odd turn, **he loses**. If the paws have no valid moves at even turn, the **King wins**. The game finishes when the King wins or loses. If the King wins, print "**King wins**" and the number of moves the King has made. If the King loses, print "**King loses**".

Some players could try to cheat by entering illegal moves, so be cautious. In case of invalid command print "Illegal move!" and enter a command again.

Example Game Session

The empty white cells are shown as "+", the empty black cells are shown as "-". The player's input is shown in *italic*:

