CS11 Programming Assignment # 4 due Thursday 21 Oct. 1999

Chess Moves

In this assignment you will write a C++ program that computes the possible destinations of a chess piece. You could think of this program as though it were part of a much larger possible project: a chess-playing program. If you were to write a complete chess-playing program, one necessary step would be to detect legal moves, which is what you will be doing in this assignment.

The main goal of this assignment is to give you a flavor of writing loops in programming, and also to reinforce the concept of modularity.

Guidelines:

1. First, ask the user to choose a rank of a chess piece. There are five alternatives: king, queen, bishop, rook and horse. The user will identify the rank by a single letter: k, q, b, r or h, respectively. (The horse is more commonly referred to as the knight, but in this program it is more convenient for each rank to begin with a different letter.) Note that you do not need to worry about pawns. Your program should detect input errors as they are encountered. If the user provides an invalid character, the program should print an informative error message and give the user another chance.

2. Second, ask the user to select the row and column in which the selected chess piece is to initially occupy. You will need to make sure the row and column numbers are within the 1..8 range. If the row number is invalid, print an error message and ask the user for the row number again before going on to the column number. Similarly, if the column number is invalid, it is only necessary to ask for the column number again (rather than requiring the user to start the whole input process over).

3. For the purpose of this program, we will assume that this piece is the *only* piece on the entire board. Your program should use loops as much as possible to find all possible legal destinations of the chess piece. Even in the cases of the king and horse, you should write loops rather than enumerating the (up to) eight destinations in your code. In your documentation you should describe in what order the possible destinations are found.

4. In chess, the king can move one square in any of the eight directions (up, down, left, right or diagonal). The bishop can move an arbitrary distance diagonally. The rook can move an arbitrary distance vertically or horizontally. The queen can go an arbitrary distance in any of the eight directions, thus encompassing the mobility of the bishop and rook. The horse can only move to a position that is both two squares in one direction and one square in a perpendicular direction, for instance from 3,5 to 5,4. In this program, a piece is not allowed to remain at its initial position. Also make sure your program does not allow a piece to go beyond the edge of the board. Recall that there are 8 rows and 8 columns in the chess board, and they are numbered 1 through 8.

5. Your program should have a different function for each rank. The queen function can be very concise since the set of possible destinations for the queen is simply the union of the destinations of the bishop and rook.

6. Note that it is not necessary to use arrays in this program. You are only printing the possible destinations within the board, not storing them for future reference.

7. The list of destinations should be prefaced by the introduction, "The possible destinations for your … are:" where the ellipsis is replaced by the name of the appropriate rank. The possible destinations of the chess piece are to be printed in the form *row*,*column* (e.g. 5,6), one set of coordinates per line (see the example output below). After your program has printed all the destinations, it should halt.

Example I/O:

```
************************************************
*       Welcome to the Chess Moves Program       *
************************************************

Which token would you like to have on the board?
Please enter [(k)ing, (q)ueen, (b)ishop, (r)ook, (h)orse]:  b
Enter starting row (1-8):  4
Enter starting column (1-8):  3
The possible destinations for your bishop are:
3,2
2,1
3,4
2,5
1,6
5,2
6,1
5,4
6,5
7,6
8,7
```