CpSc101 – Fall09 – Goddard – Notes for 11/6/09

2-Dimensional Arrays

1 2-Dimensional Arrays

A two-dimensional array can be used to represent, for example, a matrix, a table or a chess board. The row and column positions are given as successive indices. For example, a board for tictactoe might have something like

```
char board[3][3];
board[1][1] = 'X';  // the center
```

If two-dimensional arrays are passed as parameters, at least their first dimension size needs to be specified; for our examples, we will specify both sizes.

2 Sample code: chess.c

A rook in chess attacks all squares in the same row and in the same column. The following code shows where some placed rooks can attack.

```
// chess.c - wdg
// program to show a little 2D array stuff
#include <stdio.h>

void initBoard( char B[8][8] );
void placeRook( char B[8][8], int row, int col );
void printBoard( char B[8][8] );

const char CLEAR = '.';
const char ROOK = 'R';
const char ATTACKED = 'x';

int main(void) {
   char board[8][8];
   initBoard(board);
   placeRook(board,3,4);
   placeRook(board,5,0);
   printBoard(board);
   return 0;
}
```

```
// @pre: none
// @post: all entries in B are set to a period
void initBoard( char B[8][8] ) {
   int i,j;
   for(i=0; i<8; i++)
     for(j=0; j<8; j++)
        B[i][j] = CLEAR;
}

// @pre: none
// @parms: assumes 0<= row, col < 8
// @post: rook added to board B at square indicated
void placeRook( char B[8][8], int row, int col ) {
   int i;
   for(i=0; i<8; i++)
      B[row][i] = B[i][col] = ATTACKED;
   B[row][col] = ROOK;
}

// @pre: none
// @post: contents of B printed on standard output
void printBoard( char B[8][8] ) {
   int i,j;
   for(i=0; i<8; i++) {
      for(j=0; j<8; j++)
         printf("%c", B[i][j]);
    printf("\n");
  }
}
```

Practice. The placeRook function has a bug (think about rooks already on the board). It also does not validate its parameters. Revise the function to improve it.