

# Introduction to C - Program 11

## *2D Arrays and Use of Functions: Triangular Peg Game*

### Objective

To give students practice in writing functions and calling those functions to perform more complex tasks.

### The Problem: Triangular Peg Game

Many of Pizza Shack's competing restaurants have games for their customers to play while they are waiting to be seated or pick up their order. One of many people's favorites is the peg game.

You are given a triangular arrangement of 15 holes, of which 14 start with pegs. For each move, you must jump a peg over another peg into an empty hole, with a valid jump being in a straight line. The result of a jump is removing the peg that was jumped over from the board. The game ends when there are no valid jumps left. The goal is to leave as few pegs as possible.

Other restaurants have old-fashioned wood-carved playing sets of the peg game. Since Pizza Shack is trying to attract a younger clientele, your boss has asked you to write a computer program that allows customers to play the peg game electronically!

To slightly simplify your task, we assume that the open hole for the initial configuration is always in the "middle" of the design in row 3, column 2. Thus, in text, we will print out the following as the initial board configuration:

```
    P
   P P
  P . P
 P P P P
P P P P P
```

A capital 'P' represents a peg in the corresponding hole while a period represents an empty hole. We label the rows 1 through 5, and on row R, the valid columns are 1 through R, from left to right.

### Implementation Requirements

1. You must store your board in a 2 dimensional character array. (Note: No null characters need to be stored in the array since the board will always be printed character by character.)
2. You must have at least five non-trivial functions in your whole program.
3. You must have one function that takes in a board position and returns if there are any possible moves for the player from that position or not.

### Restrictions

Name the file you create and turn in *peggame.c*. Although you may use other compilers, your program must compile and run using gcc in Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. You should also include comments throughout your code, when appropriate. If you have any questions about this, please see a TA.

### Input Specification

Assume that the user always enters the valid positions for every jump. Thus, the first two values the user will enter for every prompted move will be the row and column, respectively of a hole that has a peg, using the convention previously mentioned. The second two values will represent an empty hole on the playing grid to where the jump ends and the hole in the middle of the two holes (calculated by taking the average of both the row and column coordinates of the starting and ending of the jump) are guaranteed to contain a peg. (**Note: The intention is to make the assignment easier by not requiring error checking!!!**)

### Output Specification

Before each move prompt the user with a print out of the current board, then ask them to enter in the row and column of the starting peg, followed by the row and column of the ending hole. Follow the sample program run shown below.

### Example Program Run (User input in bold and italics)

Here is the playing grid:

```
  P
   P P
  P . P
 P P P P
P P P P P
```

What are the row and column of your starting peg?

**5 4**

What are the row and column of the hole to jump to?

**3 2**

Here is the playing grid:

```
  P
   P P
  P P P
 P P . P
P P P . P
```

What are the row and column of your starting peg?

**4 1**

What are the row and column of the hole to jump to?

**4 3**

Here is the playing grid:

```
P
 P P
 P P P
 . . P P
 P P P . P
```

What are the row and column of your starting peg?

**4 4**

What are the row and column of the hole to jump to?

**4 2**

Here is the playing grid:

```
P
 P P
 P P P
 . P . .
 P P P . P
```

What are the row and column of your starting peg?

**5 2**

What are the row and column of the hole to jump to?

**5 4**

Here is the playing grid:

```
P
 P P
 P P P
 . P . .
 P . . P P
```

What are the row and column of your starting peg?

**5 5**

What are the row and column of the hole to jump to?

**5 3**

Here is the playing grid:

```
P
 P P
 P P P
 . P . .
 P . P . .
```

What are the row and column of your starting peg?

**2 2**

What are the row and column of the hole to jump to?

**4 4**

Here is the playing grid:

```
  P
  P .
 P P .
. P . P
P . P . .
```

What are the row and column of your starting peg?

**3 1**

What are the row and column of the hole to jump to?

**3 3**

Here is the playing grid:

```
  P
  P .
. . P
. P . P
P . P . .
```

What are the row and column of your starting peg?

**1 1**

What are the row and column of the hole to jump to?

**3 1**

Here is the playing grid:

```
  .
  . .
  P . P
. P . P
P . P . .
```

What are the row and column of your starting peg?

**4 4**

What are the row and column of the hole to jump to?

**2 2**

Here is the playing grid:

```
  .
  . P
  P . .
. P . .
P . P . .
```

No more jumps are possible. You have left 5 pegs.

Good bye!

### **Deliverables**

A single source file named *peggame.c* turned in through WebCourses.