

# Lab 07 - Queues Problems

**Direction:** Submit typed work in the Labs directory of your github repositor or dropbox, or upload to the google classroom assignment. The file name should be "lab7.cpp".

## Part A: In class

Your objective is to write the definition of the function `BishopMove()` whose header is

```
bool BishopMove(Vector<char>& bd,int s,int e)
```

Given that `bd` represents a chess board ( $8 \times 8$ ) that consists only of characters 'o' for free space and 'x' for occupied space, the function returns true if a bishop whose start position is `s` can make it to the end position `e` in any number of steps if `s` and `e` are both between 0 and 63 inclusively; otherwise, it returns false. It does not matter if the start and end positions characters are 'o' or 'x'. However, movement to occupied spaces are prohibited. Since `bd` is in one-dimensional array and `s` and `e` are coordinates represented in one-dimension, use the formulas

$$\begin{aligned}p(r,c) &= r * 8 + c \\ r(p) &= p / 8 \\ c(p) &= p \% 8\end{aligned}$$

where `p(r,c)` converts indices of a two-dimensional ( $8 \times 8$ ) array to an index of an equivalent one-dimensional array, `r(p)` gets the row index for a two-dimensional ( $8 \times 8$ ) array from the index of an equivalent one-dimensional array, and `c(p)` gets the column index for a two-dimensional ( $8 \times 8$ ) array from the index of an equivalent one-dimensional array. Recall that a bishop in chess can only move diagonally.

**Hint:** Keep track of positions visited.