

# Eight Queens Pretty Print

Queens College, CUNY

CS211 Lab

Instructor: Kent Chin

# typedef

*typedef* stands for **type definition**. This keyword allows us to form an alias for a datatype!

```
typedef char box[5][7];
```

*box* is now a **datatype** which refers to a **2D array, with 5 rows and 7 columns, of characters!**

```
box bb, wb, *board[8][8];
```

*bb* and *wb* are both of type *box*; that is, they are both **2D character arrays with 5 rows and 7 columns!** *board* is a **2D array, with 8 rows and 8 columns, of pointers to *box* types;** that is, each cell of *board* stores the **address** of some variable of type *box*!

# `wb` and `bb`

```
for (i = 0; i < 5; ++i)           //for each row
    for (j = 0; j < 7; ++j) { //for each column
        wb[i][j] = ' ';
        bb[i][j] = char(219);
    }
```

Remember that `wb` and `bb` are both of type `box` – that is, a 2D character array with 5 rows and 7 columns. We can treat `wb` and `bb` as arrays, **because they are arrays!**

# Filling the board

```
for (i = 0; i < 8; ++i)    //for each row
    for (j = 0; j < 8; ++j) //for each column
        if ((i+j) % 2 == 0)
            board[i][j] = &wb;
        else
            board[i][j] = &bb;
```

Remember that **pointers store memory addresses**. Each element of *board* is a **pointer** to a *box*, and therefore **stores the memory address** of a *box*. In the loops above, we alternate between white boxes and black boxes.

# How board looks

*wb*

' '	' '	' '	' '	' '	' '	' '
' '	' '	' '	' '	' '	' '	' '
' '	' '	' '	' '	' '	' '	' '
' '	' '	' '	' '	' '	' '	' '
' '	' '	' '	' '	' '	' '	' '

*bb*

'■'	'■'	'■'	'■'	'■'	'■'	'■'
'■'	'■'	'■'	'■'	'■'	'■'	'■'
'■'	'■'	'■'	'■'	'■'	'■'	'■'
'■'	'■'	'■'	'■'	'■'	'■'	'■'
'■'	'■'	'■'	'■'	'■'	'■'	'■'

&wb	&bb	&wb	&bb	&wb	&bb	&wb	&bb
&bb	&wb	&bb	&wb	&bb	&wb	&bb	&wb
&wb	&bb	&wb	&bb	&wb	&bb	&wb	&bb
&bb	&wb	&bb	&wb	&bb	&wb	&bb	&wb
&wb	&bb	&wb	&bb	&wb	&bb	&wb	&bb
&bb	&wb	&bb	&wb	&bb	&wb	&bb	&wb
&wb	&bb	&wb	&bb	&wb	&bb	&wb	&bb
&bb	&wb	&bb	&wb	&bb	&wb	&bb	&wb

We only need to make **two boxes**, *wb* and *bb*. We can then just have each cell in *board point* to either *wb* or *bb*!

# Printing the board

```
for (i = 0; i < 8; ++i) //for each board row
    for (k = 0; k < 5; ++k) { //for each box row
        cout << " " << char(179);
        for (j = 0; j < 8; ++j) //for each board column
            for (l = 0; l < 7; ++l) //for each box column
                cout << (*board[i][j])[k][l];
        cout << char(179) << endl;
    }
```

Retrieve *box pointer* from board, then **dereference**. Dereferencing a box pointer gives us the *box* being **pointed** to.

box pointer

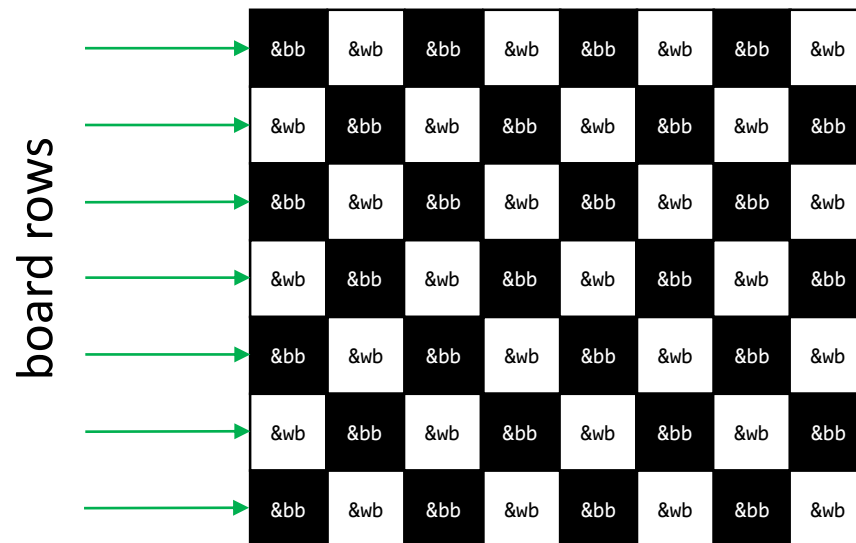
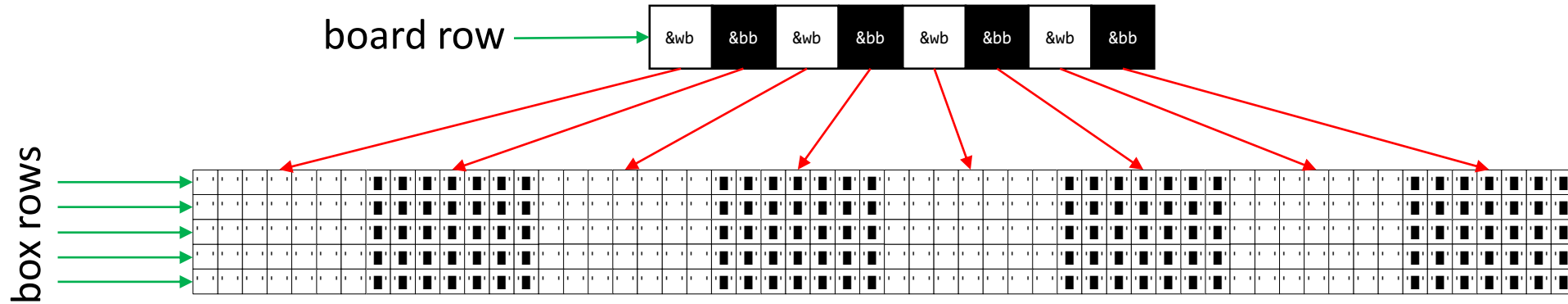
box row

box column

# Board and Box Rows

```
for (i = 0; i < 8; ++i) //for each board row
```

```
for (k = 0; k < 5; ++k) //for each box row
```



# Board and Box Columns

```
for (j = 0; j < 8; ++j) //for each board column  
    for (l = 0; l < 7; ++l) //for each box column
```

