Lab 6
Eight Queens 1D

A New Representation

\[ q[c] = r \rightarrow \text{“Queen piece placed in row } r \text{ of column } c \text{”} \]
Shell of the Eight Queens 1D Program

```c
int main() {
/*board setup section*/
next_col:
/*column section*/
next_row:
/*row section*/
backtrack:
/*backtrack section*/
print:
/*print section*/
}
```
Board Setup Section

int q[8], //1D array representation of the chess board
c = 0; //column index, initially 0
q[0] = 0; //putting 1st queen piece on upper-left corner
```c
++c; //advance column position
if (c == 8) //solutions for columns 0 through 7 found,
goto print; //so print
q[c] = -1; //otherwise start at the “top” of the column
```
Row Section

```cpp
++q[c]; //advance row position
if (q[c] == 8) //tried all rows in current column, none work,
goto backtrack; //so we backtrack
```
Row Section, continued

```plaintext
for (int i = 0; i < c; ++i) // Check up to, but not including, current column
  if (q[i] == q[c] || (c-i) == abs(q[c]-q[i]))
      goto next_row; // if conflict exists, go to next row
  goto next_col; // no conflict exists, go to next column

Note: q[i] == q[c] || (c-i) == abs(q[c]-q[i])
  // Row conflict test Up-diagonal and Down-diagonal tests
```

---

**Diagram:**

- **Row Section:**
  - Queen at row 0, column 0.
  - Queen at row 3, column 3.
  - Queen at row 4, column 6.

- **Conflict Tests:**
  - Up-diagonal: (0,0) to (3,3), (0,0) to (4,6).
  - Down-diagonal: (0,0) to (3,3), (0,0) to (4,6).

- **Next Row:**
  - Move to row 1, column 0, if conflict exists.

- **Next Column:**
  - Move to row 0, column 1, if no conflict exists.
Backtrack Section

--c; //go back one column
if (c == -1) //if past first column, all solutions found,
    return 0; //so terminate program
goto next_row; //move on to next row
print(q); //print the board

goto backtrack; //go back to find more solutions