

## Backtracking - Exercise

- Given the value 'n' as input, implement a recursive function to find all solutions to place n queens on an n×n board such that they do not attack each other.

Function Prototype: `int** nQueens(int n);`

- Implement the following variations of the nQueens Problem:
  - Given the value 'n' as input, implement a recursive function to find all solutions to place n queens on an n×n board such that they do not attack each other, ensuring that no 2 solutions are mirror images of each other.

Function Prototype: `int** nQueensNoMirrorImage(int n);`

- Given the value 'n' as input, implement a recursive function to find all solutions to place n rooks on an n×n board such that they do not attack each other. Note: A rook moves only horizontally and vertically.

Function Prototype: `int** nRooks(int n);`

- Given the value 'n' as input, implement a recursive algorithm to find all solutions to place n bishops on an n×n board such that they do not attack each other. Note: A bishop moves only diagonally.

Function Prototype: `int** nBishops(int n);`

- Consider an n×n matrix with each cell containing an alphabet of the English language. The alphabets are randomly distributed in the matrix. Given such a matrix and a list of words having meaning in the English Language, implement a function to trace the words in the matrix by starting from any position, navigating either horizontally, vertically or diagonally. The function should return the (x,y) coordinates for each letter of every word of the list found in the matrix, assuming that the bottom left most alphabet of the matrix has coordinates (0,0).

Function Prototype: `struct coordinates ** nBishops(int n);`

Sample:

```

T I B B A R N L
Z T F P X M Z E
K A V F T O I M
J M Y I L N F A
O Y G I I K T C
Z E E C O E J E
R N D Z N Y W S
    
```

Input: I D O N K E Y N , {CAMEL, DONKEY, FOX, LION, MONKEY, RABBIT, TIGER}

Output: (7,3), (7,4), (7,5), (7,6), (7,7)  
 (1,0), (2,0), (3,0), (4,0), (5,0), (6,0)  
 (6,4), (5,5), (4,6)  
 (4,4), (4,3), (4,2), (4,1)  
 (5,6), (5,5), (5,4), (5,3), (5,2), (5,1)  
 (5,7), (4,7), (3,7), (2,7), (1,7), (0,7)  
 (4,5), (3,4), (2,3), (1,2), (0,1)

```

T I B B A R N L
Z T F P X M Z E
K A V F T O I M
J M Y I L N F A
O Y G I I K T C
Z E E C O E J E
R N D Z N Y W S
    
```

Illustration: I D O N K E Y N