# N Queen Placement with Kill Order

```
def NqueenPermute(n):
    chess = [[0] * n for i in range(n)]
    queens = n
    NqueenPermuteUtil(chess, 0, -1, queens)


def NqueenPermuteUtil(chess, qsf, llb, queens):
    if qsf==queens:
        for i in range(len(chess)):
            for j in range(len(chess)):
                if chess[i][j]==1:
                    print('q',end='')
                else:
                    print('-',end='')
            print()
        print()
        return

    for i in range(llb+1, len(chess)**2):
        row = i // len(chess)
        col = i % len(chess)
        if chess[row][col]==0 and isSafe(chess,row,col):
            chess[row][col]=1
            NqueenPermuteUtil(chess, qsf+1, row*len(chess)+col, queens)
            chess[row][col]=0

def isSafe(chess, row, col):
    i = row - 1
    while i >= 0:
        if chess[i][col] == 1:
            return False
        i = i - 1

    i = row - 1
    j = col - 1
    while i >= 0 and j >= 0:
        if chess[i][j] == 1:
            return False
        i = i - 1
```

```
        j = j - 1
    i = row - 1
    j = col + 1
    while i >= 0 and j < len(chess):
        if chess[i][j] == 1:
            return False
        i = i - 1
        j = j + 1
    i = row
    j = col-1
    while j>=0:

        if chess[i][j]==1:
            return False
        j = j-1

    return True



NqueenPermute(4)
```

Here queen chooses.