In [5]:
```
def queen_safe(row, qsofar):
    """ returns True if it is safe to place another
    queen in `row` in column `len(qsofar)`, given
    existing placement of queens in `qsofar` in the first
    `len(qsofar)` columns """
    col = len(qsofar)
    for (pcol,prow) in enumerate(qsofar):
        # can't place queen in same row
        if prow == row:
            return False
        # diagonal attack
        if abs(prow-row) == col-pcol:
            return False
    # no problems with any existing queens
    return True
```

In [6]:
```
queen_safe(4,[0,2])
```

Out[6]: True

In [10]:
```
def nqueens(n, qsofar):
    col = len(qsofar)
    if col == n:
        return True
    count = 0
    for row in range(n):
        if queen_safe(row, qsofar):
            print(f"Placing new queen {qsofar} {row}")
            if nqueens(n, qsofar + [row]):
                return True
    return False
```

In [11]:
```
def nqueens_count(n, qsofar):
    col = len(qsofar)
    if col == n:
        return 1
    count = 0
    for row in range(n):
        if queen_safe(row, qsofar):
            #print(f"Placing new queen {qsofar} {row}")
            count += nqueens_count(n, qsofar + [row])
    return count
```

In [12]:
```
nqueens(8,[])
```

```
Placing new queen [] 0
```

```
Placing new queen [0] 2
Placing new queen [0, 2] 4
Placing new queen [0, 2, 4] 1
Placing new queen [0, 2, 4, 1] 3
Placing new queen [0, 2, 4, 1] 7
Placing new queen [0, 2, 4] 6
Placing new queen [0, 2, 4, 6] 1
Placing new queen [0, 2, 4, 6, 1] 3
Placing new queen [0, 2, 4, 6, 1, 3] 5
Placing new queen [0, 2, 4, 6] 3
Placing new queen [0, 2, 4] 7
Placing new queen [0, 2, 4, 7] 1
Placing new queen [0, 2, 4, 7, 1] 3
Placing new queen [0, 2, 4, 7, 1, 3] 5
Placing new queen [0, 2, 4, 7] 3
Placing new queen [0, 2] 5
Placing new queen [0, 2, 5] 1
Placing new queen [0, 2, 5, 1] 6
Placing new queen [0, 2, 5, 1, 6] 4
Placing new queen [0, 2, 5] 7
Placing new queen [0, 2, 5, 7] 1
Placing new queen [0, 2, 5, 7, 1] 3
Placing new queen [0, 2, 5, 7, 1] 4
Placing new queen [0, 2] 6
Placing new queen [0, 2, 6] 1
Placing new queen [0, 2, 6, 1] 3
Placing new queen [0, 2, 6, 1, 3] 7
Placing new queen [0, 2, 6, 1] 7
Placing new queen [0, 2, 6, 1, 7] 4
Placing new queen [0, 2] 7
Placing new queen [0, 2, 7] 1
Placing new queen [0, 2, 7, 1] 3
Placing new queen [0, 2, 7, 1] 6
Placing new queen [0, 2, 7] 5
Placing new queen [0, 2, 7, 5] 1
Placing new queen [0, 2, 7, 5] 3
Placing new queen [0, 2, 7, 5, 3] 1
Placing new queen [0, 2, 7, 5, 3, 1] 4
Placing new queen [0] 3
Placing new queen [0, 3] 1
Placing new queen [0, 3, 1] 4
Placing new queen [0, 3, 1, 4] 2
Placing new queen [0, 3, 1, 4] 7
Placing new queen [0, 3, 1] 6
Placing new queen [0, 3, 1, 6] 2
Placing new queen [0, 3, 1] 7
Placing new queen [0, 3, 1, 7] 2
Placing new queen [0, 3, 1, 7, 2] 6
Placing new queen [0, 3, 1, 7] 5
Placing new queen [0, 3, 1, 7, 5] 2
```

```
Placing new queen [0, 3] 5
Placing new queen [0, 3, 5] 2
Placing new queen [0, 3, 5] 7
Placing new queen [0, 3, 5, 7] 1
Placing new queen [0, 3, 5, 7, 1] 4
Placing new queen [0, 3, 5, 7, 1, 4] 2
Placing new queen [0, 3, 5, 7, 1] 6
Placing new queen [0, 3, 5, 7, 1, 6] 2
Placing new queen [0, 3, 5, 7] 2
Placing new queen [0, 3, 5, 7, 2] 4
Placing new queen [0, 3, 5, 7, 2] 6
Placing new queen [0, 3] 6
Placing new queen [0, 3, 6] 2
Placing new queen [0, 3, 6, 2] 5
Placing new queen [0, 3, 6, 2, 5] 1
Placing new queen [0, 3, 6, 2, 5, 1] 4
Placing new queen [0, 3, 6, 2] 7
Placing new queen [0, 3, 6, 2, 7] 1
Placing new queen [0, 3, 6, 2, 7, 1] 4
Placing new queen [0, 3, 6] 4
Placing new queen [0, 3, 6, 4] 1
Placing new queen [0, 3, 6, 4] 2
Placing new queen [0, 3, 6, 4] 7
Placing new queen [0, 3, 6, 4, 7] 1
Placing new queen [0, 3] 7
Placing new queen [0, 3, 7] 2
Placing new queen [0, 3, 7] 4
Placing new queen [0, 3, 7, 4] 1
Placing new queen [0, 3, 7, 4] 2
Placing new queen [0] 4
Placing new queen [0, 4] 1
Placing new queen [0, 4, 1] 5
Placing new queen [0, 4, 1, 5] 2
Placing new queen [0, 4, 1, 5, 2] 6
Placing new queen [0, 4, 1, 5, 2, 6] 3
Placing new queen [0, 4, 1] 7
Placing new queen [0, 4, 1, 7] 2
Placing new queen [0, 4, 1, 7, 2] 6
Placing new queen [0, 4, 1, 7, 2, 6] 3
Placing new queen [0, 4, 1, 7] 5
Placing new queen [0, 4, 1, 7, 5] 2
Placing new queen [0, 4, 1, 7, 5] 3
Placing new queen [0, 4] 6
Placing new queen [0, 4, 6] 1
Placing new queen [0, 4, 6, 1] 3
Placing new queen [0, 4, 6, 1, 3] 7
Placing new queen [0, 4, 6, 1] 5
Placing new queen [0, 4, 6, 1, 5] 2
Placing new queen [0, 4, 6, 1, 5] 7
Placing new queen [0, 4] 7
```

```
Placing new queen [0, 4, 7] 1
Placing new queen [0, 4, 7, 1] 3
Placing new queen [0, 4, 7, 1, 3] 6
Placing new queen [0, 4, 7, 1, 3, 6] 2
Placing new queen [0, 4, 7, 1] 6
Placing new queen [0, 4, 7, 1, 6] 2
Placing new queen [0, 4, 7, 1, 6, 2] 5
Placing new queen [0, 4, 7] 5
Placing new queen [0, 4, 7, 5] 2
Placing new queen [0, 4, 7, 5, 2] 6
Placing new queen [0, 4, 7, 5, 2, 6] 1
Placing new queen [0, 4, 7, 5, 2, 6, 1] 3
```

Out[12]: True

In [13]:
```
nqueens_count(8,[])
```

Out[13]: 92

In [14]:
```
def subset_sum(lst, target):
    print(f"{lst} {target}")
    if target < 0:
        return False
    if target == 0:
        return True
    if lst == []:
        return False
    return subset_sum(lst[1:], target-lst[0]) or \
        subset_sum(lst[1:], target)
```

In [15]:
```
subset_sum([1,1,5,5,10,25], 26)
```

```
[1, 1, 5, 5, 10, 25] 26
[1, 5, 5, 10, 25] 25
[5, 5, 10, 25] 24
[5, 10, 25] 19
[10, 25] 14
[25] 4
[] -21
[] 4
[25] 14
[] -11
[] 14
[10, 25] 19
[25] 9
[] -16
[] 9
[25] 19
[] -6
```

```
[] 19
[5, 10, 25] 24
[10, 25] 19
[25] 9
[] -16
[] 9
[25] 19
[] -6
[] 19
[10, 25] 24
[25] 14
[] -11
[] 14
[25] 24
[] -1
[] 24
[5, 5, 10, 25] 25
[5, 10, 25] 20
[10, 25] 15
[25] 5
[] -20
[] 5
[25] 15
[] -10
[] 15
[10, 25] 20
[25] 10
[] -15
[] 10
[25] 20
[] -5
[] 20
[5, 10, 25] 25
[10, 25] 20
[25] 10
[] -15
[] 10
[25] 20
[] -5
[] 20
[10, 25] 25
[25] 15
[] -10
[] 15
[25] 25
[] 0
```

Out[15]: True

In [16]:
```
subset_sum([1,1,5,5,10,25], 28)
```

```
[1, 1, 5, 5, 10, 25] 28
[1, 5, 5, 10, 25] 27
[5, 5, 10, 25] 26
[5, 10, 25] 21
[10, 25] 16
[25] 6
[] -19
[] 6
[25] 16
[] -9
[] 16
[10, 25] 21
[25] 11
[] -14
[] 11
[25] 21
[] -4
[] 21
[5, 10, 25] 26
[10, 25] 21
[25] 11
[] -14
[] 11
[25] 21
[] -4
[] 21
[10, 25] 26
[25] 16
[] -9
[] 16
[25] 26
[] 1
[] 26
[5, 5, 10, 25] 27
[5, 10, 25] 22
[10, 25] 17
[25] 7
[] -18
[] 7
[25] 17
[] -8
[] 17
[10, 25] 22
[25] 12
[] -13
[] 12
[25] 22
```

```
[] -3
[] 22
[5, 10, 25] 27
[10, 25] 22
[25] 12
[] -13
[] 12
[25] 22
[] -3
[] 22
[10, 25] 27
[25] 17
[] -8
[] 17
[25] 27
[] 2
[] 27
[1, 5, 5, 10, 25] 28
[5, 5, 10, 25] 27
[5, 10, 25] 22
[10, 25] 17
[25] 7
[] -18
[] 7
[25] 17
[] -8
[] 17
[10, 25] 22
[25] 12
[] -13
[] 12
[25] 22
[] -3
[] 22
[5, 10, 25] 27
[10, 25] 22
[25] 12
[] -13
[] 12
[25] 22
[] -3
[] 22
[10, 25] 27
[25] 17
[] -8
[] 17
[25] 27
[] 2
[] 27
[5, 5, 10, 25] 28
```

```
[5, 10, 25] 23
[10, 25] 18
[25] 8
[] -17
[] 8
[25] 18
[] -7
[] 18
[10, 25] 23
[25] 13
[] -12
[] 13
[25] 23
[] -2
[] 23
[5, 10, 25] 28
[10, 25] 23
[25] 13
[] -12
[] 13
[25] 23
[] -2
[] 23
[10, 25] 28
[25] 18
[] -7
[] 18
[25] 28
[] 3
[] 28
```

Out[16]: False