

```

#define ALLOC_LENGTH          (100)

bool checkNQueens(int n, int* pos, int count)
{
    int row_index;
    int diag_count;

    diag_count = 1;
    row_index = count-1;

    while(row_index >= 0)
    {
        if( ( pos[row_index] == pos[count] ) ||
            ( ( pos[count]-diag_count) >= 0 ) && ( pos[row_index] == (pos[count]-diag_count) ) ) ||
            ( ( pos[count]+diag_count) <= (n-1) ) && ( pos[row_index] == (pos[count]+diag_count) ) )
        )
        {
            return false;
        }

        diag_count++;
        row_index--;
    }

    return true;
}

void _totalNQueens(int n, int* pos, int count, int* returnSize) {
    int index;

    for(index = 0; index < n; index++)
    {
        pos[count] = index;

        if( checkNQueens(n, pos, count) )
        {
            if(count >= (n-1))
            {
                (*returnSize)++;
            }else
            {
                _totalNQueens(n, pos, count+1, returnSize);
            }
        }
    }
}

int totalNQueens(int n) {
    int* pos;
    int index;
    int returnSize;

    pos = (int*)malloc(sizeof(int) *n);

    returnSize = 0;

    _totalNQueens(n, pos, 0, &returnSize);

    return returnSize;
}

```