

## 3) (10 pts) DSN (Backtracking)

Consider the problem of placing 8 kings on an 8 x 8 chessboard, so that no two of the kings can attack each other AND no two kings are on the same row or column. (Recall that a King can move one space in each of the eight possible directions of movement: up, down, left, right or any of the four diagonals.) Complete the code skeleton below so that it prints out each solution to the 8 Kings problem. (Note: assume that the function print, which isn't included, prints out the solution that corresponds to a particular permutation of kings. For example, the permutation {2, 4, 6, 1, 3, 5, 7, 0} represents kings at the following locations (0, 2), (1, 4), (2, 6), (3, 1), (4, 3), (5, 5), (6, 7), and (7, 0).)

```
#include <stdio.h>
#include <math.h>
#define SIZE 8

void go(int perm[], int k, int used[]);
void print(int perm[]);

int main() {
    int perm[SIZE];
    int used[SIZE];
    int i;
    for (i=0; i<SIZE; i++) used[i] = 0;
    go(perm, 0, used);
    return 0;
}

void go(int perm[], int k, int used[]) {

    if ( k == SIZE ) {                //(1 pt)
        print(perm);
        return;
    }

    int i;
    for (i=0; i<SIZE; i++) {

        if ( k > 0 && abs(i-perm[k-1]) <= 1 ) continue; //(4 pts)

        if ( used[i] ) continue;      //(1 pt)

        perm[k] = i ;                //(1 pt)
        used[ i ] = 1 ;              //(1 pt)

        go(perm, k+1 , used); //(1 pt)

        used[ i ] = 0 ; //(1 pt)
    }
}
```