1) (10 pts) DSN (Recursive Coding)

A derangement is a permutation of the integers 1, 2, 3, ..., n such that for all i, $1 \leq i \leq n$, the value in the i^{th} location isn't i. For example, (2, 1, 4, 3) is a derangement of 4 items since the first item isn't 1, the second item isn't 2, the third item isn't 3 and the fourth item isn't 4. But (3, 1, 5, 4, 2) is NOT a derangement of 5 items since the 4th item on this list is 4. Complete the code below so that it prints out all derangements of size n ($2 \leq n \leq 10$), where n is entered by the user.

```
#include <stdio.h>
#define MAX 10

void printD(int n);
void printDRec(int n, int* perm, int* used, int k);
void print(int* perm, int length);

int main() {
    int n;
    printf("Enter the size of your derangement(2-10).\n");
    scanf("%d", &n);
    printD(n);
    return 0;
}
void printD(int n) {
    int perm[MAX];
    int used[MAX];
    int i;
    for (i=0; i<MAX; i++) used[i] = 0;
    printDRec( n, perm, used, 0); // Grading: 1 pt
}

void printDRec(int n, int* perm, int* used, int k) {
    if (k == n) {
        print(perm, n);
        return;
    }

    int i;
    for (i=0; i<n; i++) {
        if (!used[i] && k != i ) {                  // Grading: 3 pts

            perm[ k ] =  i ;        // Grading: 2 pts, exchange i,k okay.

            used[ i ] =  1 ;                        // Grading: 2 pts
            printDRec(n, perm, used, k+1);

            used[ i ] = 0 ;                         // Grading: 2 pts
        }
    }
}

void print(int* perm, int length) {
    int i;
    for (i=0; i<length; i++)
        printf("%d ", perm[i]+1);
    printf("\n");
}
```