1) (10 pts) DSN (Recursive Coding)

Define an extreme permutation of the integers 0 to n – 1 as any permutation where one where each value in the permutation (from left to right) is either the smallest or largest value not yet placed. For example, for n = 6, [0, 1, 5, 2, 3, 4] is an extreme permutation but [0, 5, 2, 4, 1, 3] is not. The latter is not because the only valid values that can be placed where the 2 is are either 1 or 4, the smallest and largest values, respectively, that have not been placed. Complete the recursive function below so that it prints out all extreme permutations of length n. A completed wrapper function has been provided. Note: low represents the lowest unplaced value, high represents the highest unplaced value, and k represents the number of items in the permutation that have already been filled.

```
#include <stdio.h>
#include <stdlib.h>

void printExtremeWrapper(int n);
void printExtreme(int* perm, int n, int low, int high, int k);
void printPerm(int* perm, int n);

void printExtremeWrapper(int n) {
    int* perm = malloc(sizeof(int)*n);
    printExtreme(perm, n, 0, n-1, 0);
    free(perm);
}

void printPerm(int* perm, int n) {
    for (int i=0; i<n; i++) printf("%d, ", perm[i]);
    printf("\n");
}
void printExtreme(int* perm, int n, int low, int high, int k) {

    if (low > high) {
        printPerm(perm, n);
        return;
    }


    perm[k] = low;                                  // Grading 2 pts
    printExtreme(perm, n, low+1, high, k+1);        // Grading 2 pts

    if (low == high) return;                        // Grading 2 pts

    perm[k] = high;                                 // Grading 2 pts
    printExtreme(perm, n, low, high-1, k+1);        // Grading 2 pts

}
```

Grading notes: It's possible students use a loop to try all possible values in slot k. If they write regular permutation code (so print all perms), then award 4/10 points. If students write the loop but only recurse if the value they put into the slot is low or high, give full credit (this works). Note that no order is imposed by the question so students can try high before low. Give 8 out of 10 if they repeat that recursive call twice (ie. always call with both low and high.)