3) (10 pts) DSN (Backtracking)

A “unique” positive integer of n digits is such that no two adjacent digits differ by less than 2. Specifically, given an n digit number, $d_0d_1\ldots d_{n-1}$, where d_0 is the most significant digit, (and thus, this one digit can’t be 0), $|d_i - d_{i+1}| \geq 2$ for all i ($0 \leq i \leq n-2$). Consider the problem of printing out all “unique” positive integers of n digits via backtracking, in numerical order. Fill in the code below to complete the task. (To run the code, one would have to call printWrapper with their desired parameter.)

```
#include <stdio.h>
#include <math.h>
void print(int number[], int n);
void printWrapper(int n);
void printRec(int number[], int k, int n);

void printWrapper(int n) {
    int* array = malloc(sizeof(int)*n);
    printRec(array, 0, n);
    free(array);
}

void printRec(int number[], int k, int n) {
    if (k == n) {
        print(number, n) ;                      // Grading: 1 pt

        return ;                                // Grading: 1 pt
    }

    int start = 0;

    if ( k==0 )                                 // Grading: 1 pt

        start = 1 ;                             // Grading: 1 pt
    for (int i=start; i < 10 ; i++) {           // Grading: 1 pt

        if (k > 0 && abs(number[k-1]-i)<2 )    // Grading: 2 pts
            continue;

        number[ k ] = i ;                       // Grading: 2 pts

        printRec(number, k+1, n) ;              // Grading: 1 pt
    }
}

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