3) (10 pts) DSN (Backtracking)

Consider printing out all strings of x A's and y B's, where x ≥ y-1 such that no two consecutive letters are Bs, in alphabetical order. For example, if x = 5 and y = 3, one of the valid strings printed would be AABABABA. One way to solve this problem would be to use backtracking, where a string is built up, letter by letter (first trying A, then trying B in the current slot). **Complete the code below to implement this backtracking solution idea.** The correct condition for when you can place As is already in the code. (Hint: You can only place Bs if there are Bs left to place. If there are Bs left, then you must ensure that if there is a previous letter, it is not a B.)

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

void printAll(char buffer[], int k, int a, int b);
void printWrapper(int x, int y);

// Prints all strings with exactly x A's and y B's in alphabetical
// order.
void printWrapper(int x, int y) {
    char* buffer = malloc(sizeof(char)*(x+y+1));
    buffer[x+y] = '\0';
    printAll(buffer, 0, x, y);
    free(buffer);
}

void printAll(char buffer[], int k, int x, int y) {

    if (x == 0 && y == 0) {
        printf("%s\n", buffer);
        return;
    }

    if (x > y-1) {
        buffer[k] = 'A' ;

        printAll(buffer, _____ , _____ , _____ );
    }

    if ( _____ && ( _____ || (_____ && _____) ) ){

        buffer[k] = 'B' ;

        printAll(buffer, _____ , _____ , _____ );
    }
}
```