Generalized Towers of Hanoi

Define a recursive function:

```
     void tgen(int n,  char* i,  char* f)
     {

     }
```

n is the total number of disks.

i is the initial configuration.

f is the final configuration.

tgen prints the moves to transform the initial configuration into the final configuration.

A configuration is described by a char array of tower letters, A, B or C, listed largest disk to smallest.

Because we have the disk count n, the char arrays representing the configuration do not need a terminating null, and can have extra characters after the n characters that describe the configuration.

One advantage of keeping track of n instead of marking the end of the strings with null, is that when we need a string representing all the disks on the same tower, we can use an array with more characters than are actually needed.

When you need to pass in a configuration with the largest disk removed, use pointer arithmetic. If c is a configuration, c + 1 is the configuration with the largest disk removed.

Also, when you need a configuration with all disks on the same tower, use a pointer to a pre-made array of characters. We will discuss in class how to do this.

Write a main program that accepts two configurations from the user and prints out the moves to transform one configuration into the other.