Part I: Write and Test C code/program

Move the disks from peg A to peg C, one disk at a time without a larger disk being on top of a smaller disk. Assume disk 4 is the green bottom disk, disk 3 is the red disk, disk 2 is the yellow disk, and disk 1 is the blue disk. Check out this game page to see the steps: https://www.mathsisfun.com/games/towerofhanoi.html

Implement the following two alternatives and trace them to see which works and how many steps it takes.

//alternative approach 1:
void towers(int n, char cFromPeg, char cAuxPeg, char cToPeg){
    // only one disk to move
    if (n == 1){
        printf("move disk %d from %c to %c\n", n, cFromPeg, cToPeg);
        return;
    }
    // move the top n-1 disks to the auxPeg using the toPeg as an aux
    towers(n-1, cFromPeg, cToPeg, cAuxPeg);

    // move disk n
    printf("move disk %d from %c to %c\n", n, cFromPeg, cToPeg);

    // move the top n-1 disks from auxPeg to toPeg using fromPeg as an aux
    towers(n-1, cAuxPeg, cFromPeg, cToPeg);
}

//alternative approach 2:
void towers(int n, char cFromPeg, char cAuxPeg, char cToPeg){
    // only one disk to move
    if (n == 1){
        printf("move disk %d from %c to %c\n", n, cFromPeg, cToPeg);
        return;
    }
    // move the top n-1 disks to the auxPeg using the toPeg as an aux
    towers(n-1, cFromPeg, cToPeg, cAuxPeg);

    // move disk n
    printf("move disk %d from %c to %c\n", n, cFromPeg, cToPeg);

    // move the top n-1 disks from auxPeg to fromPeg using toPeg as an aux
    towers(n-1, cAuxPeg, cToPeg, cFromPeg);
}