Solutions to the Banish Winter Problem

/*
 * File: BanishWinter.k
 * -----------
 * The BanishWinter program gets Karel adorn a series of trees with a
 * cluster of beeper leaves.
 */

import "turns";

/*/ Main function */

function banishWinter() {
    while (beepersInBag()) {
        findTree();
        addLeavesToTree();
    }
}

 acts Karel up to the next tree.
* *
 * Programming style note: Since a tree is simply a wall, this method
 * can simply call moveToWall. You could therefore replace the
 * findTree call in the main program with moveToWall, but the program
 * might then be harder to read because it violates the "tree" metaphor
 * used at the level of the main program.
 */

function findTree() {
    moveToWall();
}

 acts Karel up to the next tree. The precondition
 * is that Karel must be immediately west of the tree, facing east;
 * the postcondition is that Karel is at the bottom of the other side
 * of the tree, facing east.
 */

function addLeavesToTree() {
    turnLeft();
    climbTree();
    addLeaves();
    descendToGround();
    turnLeft();
}
function climbTree() {
    while (rightIsBlocked()) {
        move();
    }
}

function descendToGround() {
    moveToWall();
}

function addLeaves() {
    turnRight();
    makeBeeperSquare();
    move();
    turnRight();
}

function moveToWall() {
    while (frontIsClear()) {
        move();
    }
}

function makeBeeperSquare() {
    repeat (4) {
        putBeeper();
        move();
        turnLeft();
    }
}