Solutions to Class Exercises

This handout was written by Eric Roberts.

```c
/*
 * File: PutBeepLine.k
 * ---------------
 * This program implements and tests the PutBeepLine function.
 */

/*
 * Implements the simple test function used in class.
 */

function test() {
    putBeepLine();
    turnLeft();
    putBeepLine();
}

/*
 * Puts down a line of beepers up to and including the wall.
 */

function putBeepLine() {
    while (frontIsClear()) {
        putBeeper();
        move();
    }
    putBeeper();
}
```
/*
 * File: MountainKarel1.k
 * 
 * This program instructs Karel to climb a simple mountain, plant a
 * flag, and descend to the ground. This version works only for the
 * specific world shown in the handout.
 */

import "turns";

/*
 * Climbs the specific mountain shown in the handout.
 */

function climbMountain() {
    moveToWall();
    turnLeft();
    move();
    turnRight();
    move();
    turnLeft();
    move();
    turnRight();
    move();
    turnLeft();
    move();
    turnRight();
    move();
    putBeeper();
    move();
    turnRight();
    move();
    turnLeft();
    move();
    turnRight();
    move();
    turnLeft();
    move();
    turnRight();
    move();
    turnLeft();
    move();
    turnRight();
    move();
    turnLeft();
    move();
    moveToWall();
}

/*
 * Moves Karel forward until it is blocked by a wall.
 */

function moveToWall() {
    while (frontIsClear()) {
        move();
    }
}
/*
 * File: MountainKarel2.k
 * ---------------------
 * This program is the same as MountainKarel1 but defines the functions
 * stepUp and stepDown to simplify the code.
 */

import "turns";

/*
 * Climbs the specific mountain shown in the handout.
 */

function climbMountain() {
    moveToWall();
    stepUp();
    stepUp();
    stepUp();
    putBeeper();
    stepDown();
    stepDown();
    stepDown();
    stepDown();
    moveToWall();
}

/*
 * Sends Karel up the step ahead of it.
 */

function stepUp() {
    turnLeft();
    move();
    turnRight();
    move();
}

/*
 * Sends Karel down the step ahead of it.
 */

function stepDown() {
    move();
    turnRight();
    move();
    turnLeft();
}

/* The moveToWall function is the same as in the earlier programs */
/ * File: MountainKarel3.k
 * ---------------------
 * This program again solves only the mountain world from the handout,
 * but does so using the repeat statement in a way that makes it easy
 * to adapt the program to climb a stair-step mountain of any size.
 */

import "turns";

/*
 * Climbs a stair-step mountain whose size appears in the repeat statements.
 */

function climbMountain() {
  moveToWall();
  repeat (3) {
    stepUp();
  }
  putBeeper();
  repeat (3) {
    stepDown();
  }
  moveToWall();
}

/*
 * Sends Karel up the step ahead of it.
 */

function stepUp() {
  turnLeft();
  move();
  turnRight();
  move();
}

/*
 * Send Karel down the step ahead of it.
 */

function stepDown() {
  move();
  turnRight();
  move();
  turnLeft();
}

/* The moveToWall function is the same as in the earlier programs */
/* File: MountainKarel4.k */
*
* This version of the program attempts to generalize the solution
* strategy so that it climbs a stair-step mountain of any size, but
* it fails if the end of the world appears immediately after the
* base of the mountain.
*/

import "turns";

/*
* Tries to climb a stair-step mountain of any size.
*/

function climbMountain() {
    moveToWall();
    while (frontIsBlocked()) {
        stepUp();
    }
    putBeeper();
    move();
    while (rightIsClear()) {
        dropDown();
    }
    moveToWall();
}

/*
* Sends Karel up the step ahead of it.
*/

function stepUp() {
    turnLeft();
    move();
    turnRight();
    move();
}

/*
* Drops down from the midair position just past a descending step.
*/

function dropDown() {
    turnRight();
    move();
    turnLeft();
    move();
}

/* The moveToWall function is the same as in the earlier programs */
/* File: MountainKarel5.k
 * ---------------------
 * This version fixes the bug in MountainKarel4.k so that it
 * checks for a wall before moving forward in dropDown.
 */

import "turns";

/*
 * Climbs a stair-step mountain of any size.
 */

function climbMountain() {
    moveToWall();
    while (frontIsBlocked()) {
        stepUp();
    }
    putBeeper();
    move();
    while (rightIsClear()) {
        dropDown();
    }
    moveToWall();
}

/*
 * Sends Karel up the step ahead of it.
 */

function stepUp() {
    turnLeft();
    move();
    turnRight();
    move();
}

/*
 * Drops down from the midair position just past a descending step.
 */

function dropDown() {
    turnRight();
    move();
    turnLeft();
    if (frontIsClear()) {
        move();
    }
}

/* The moveToWall function is the same as in the earlier programs */