Section Handout #1

This week in section, your first priority is to meet your section leader and discover what sections in CS106A are all about. Your section leader will therefore spend the first part of this week's session on introductions and telling you the things you need to know, such as where to get help and how to sign up for interactive grading. After the introductory material, the section will move on to cover some of the important material from class in a setting that is small enough for you to ask questions and thereby find out what you need to know. Make sure that you get your section leader's email address in section, so that you can complete the Email portion of Assignment #1! This week, your goal is to solve a Karel problem that involves stepwise refinement:

Karel Defends Democracy!

In the 2000 US presidential elections, most votes were cast on paper ballots where voters would punch out a section of the ballot to mark their preference. That way, votes could easily be tallied by hand or by machine by checking for holes in the ballots. Unfortunately, though, this system was prone to a problem called the *hanging chad* in which a voter would attempt to punch a hole in the ballot without completely creating a hole. The paper that was intended to be punched out would then be hanging off the back of the ballot, making it possible for the vote to accidentally be miscounted.

Since then, most states eliminated the punch-card ballot problems that made hanging chad a household phrase at the end of 2000. The most common replacement is electronic voting machines—which have problems of their own.

Another possible (if fanciful) strategy for securing elections would be to retain the punch cards but to have a miniaturized robot—Karel, of course—check each of the ballots before it is counted to ensure that no unwanted chad remains. Karel's job is to move across the punch card ballot and make sure that no stray bits of the card remain in any of the ballot spaces in which the user has attempted to cast a vote.

To make this task more concrete, imagine that Karel is sitting at the extreme left edge of a punch-card ballot that looks like this:
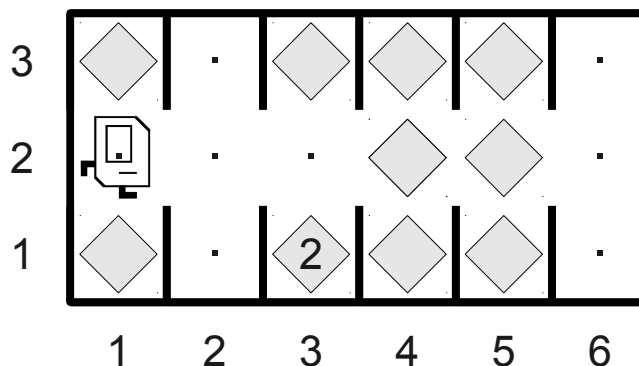
The partially enclosed rectangles in the interior of the card represent the areas of the ballot that voters must punch out to record their preferences. On the original ballot, these rectangles are completely

filled with beepers. Ideally, the voter will punch out all the beepers when casting a vote, leading to an empty rectangle of the sort shown on 2nd avenue. Unfortunately, some bits of the card—the chad—sometimes end up remaining in the hole, as shown on 1st and 3rd avenues, where although the middle square is punched out, beepers still remain in the ballot.

Suppose that your state legislature has determined that the voter's intent is indicated by the status of the square in the middle of the rectangle, which is where the stylus makes contact with the card. If there is a beeper in that position, Karel must assume that the voter did not intend to cast a vote in that column and move on to the next. If there is no beeper in the center square, Karel must check the other two squares in the ballot and remove any and all beepers so that the ballot can be counted correctly. Thus, the final configuration of the ballot after Karel completes the processing should look like this:
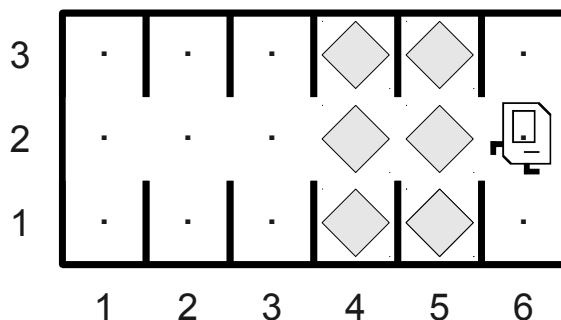
Karel may count on the following facts about the world:

- The world consists of a single row of ballot rectangles that appear on every avenue, as shown in the sample world. The size of the ballot, however, may be different from the one shown in the example in the sense that it may contain any number of ballot rectangles. In any case, the card is completely made from ballot rectangles, with no buffer to the left or right.

- Every ballot rectangle is exactly one space wide and three spaces high, as shown in the diagram.

- Karel always begins inside of the leftmost ballot rectangle, facing the hole that gives Karel access to the voting area along the center line of the rectangles.

- Karel must finish execution facing east at the rightmost edge of the ballot.

Write a Karel program to clean the chad from a ballot. Remember that your program should not work only for the example shown in the diagram, but for any ballot that meets these conditions.