

# ATYPON

## Karel Assignment

Made by: Samer AlHamdan

Instructors:

Motasim Aldiab

Fahed Jubair

<b>1. Introduction</b>	<b>3</b>
<b>2. The Problem</b>	<b>3</b>
2.1 Description	3
2.2 Finding the Optimal Solutions	3
<b>3. Solution</b>	<b>5</b>
3.1 Calculating the Dimensions	5
3.2 Drawing the Outer Chambers	6
3.3 Drawing the Inner Chambers	8
<b>4. Overall Algorithm</b>	<b>9</b>

# 1. Introduction

This report offers a comprehensive analysis of the issue identified in the Karel Assignment during Atypen's Training in May 2023. It explores a range of potential solutions, evaluates them, and ultimately recommends the most effective solution while striving for optimization. Additionally, the report Highlights the optimizations made within the code to achieve the optimal solution.

## 2. The Problem

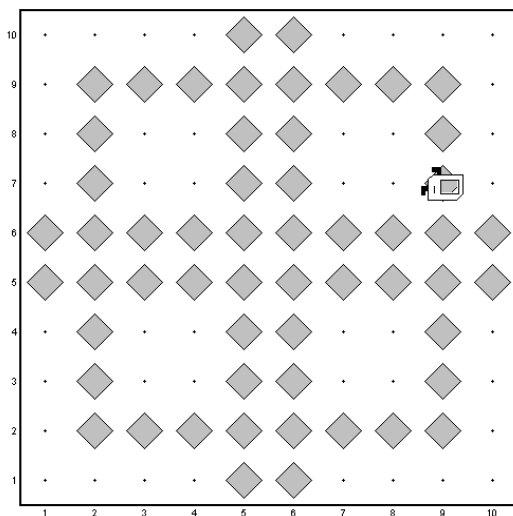
### 2.1 Description

The problem addressed by Karel's Assignment is the need to divide a given map of unknown dimensions ( $N \times M$ ) into 4 + 4 chambers (if applicable). The 4 inner chambers should be the biggest possible equal squares, The 4 outer chambers should be L-Shaped and equal in size, in the lowest number of moves, and the lowest number of lines in code. If dividing the map into 4 + 4 chambers is not applicable, Karel won't be able to work with the dimensions that have been given to him.

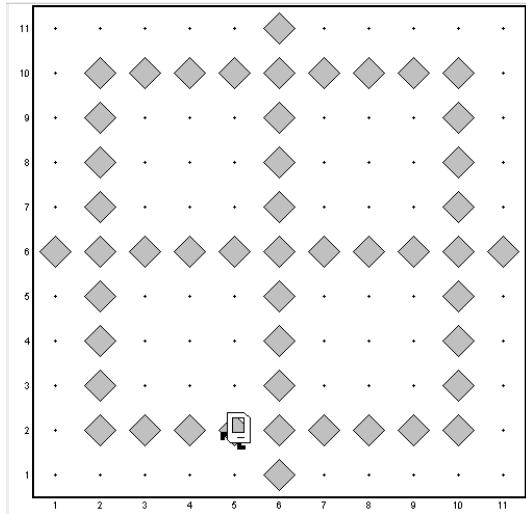
### 2.2 Finding the Optimal Solutions

Having multiple valid ways to divide the map into 4 + 4 chambers, I had to find the best way that gives me the least moves and the least beepers. Below are my optimal solutions to all possible cases.

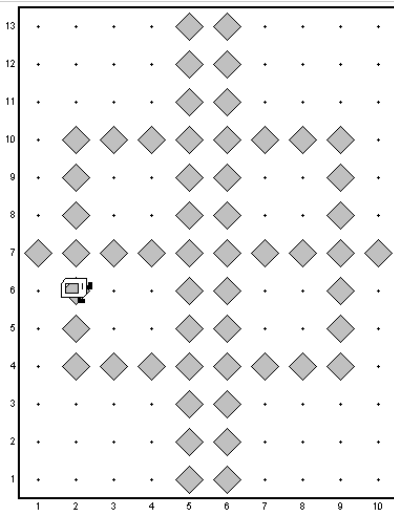
Case #1 Even rows Even columns:



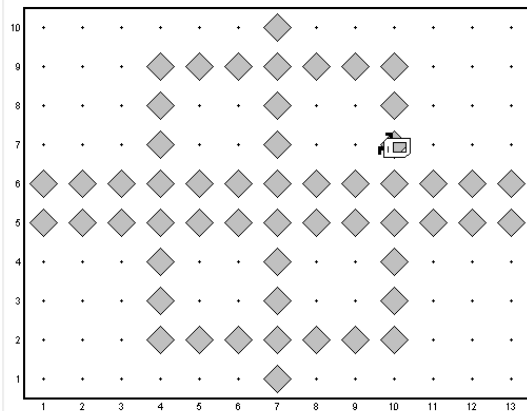
Case #2 Odd rows Odd columns:



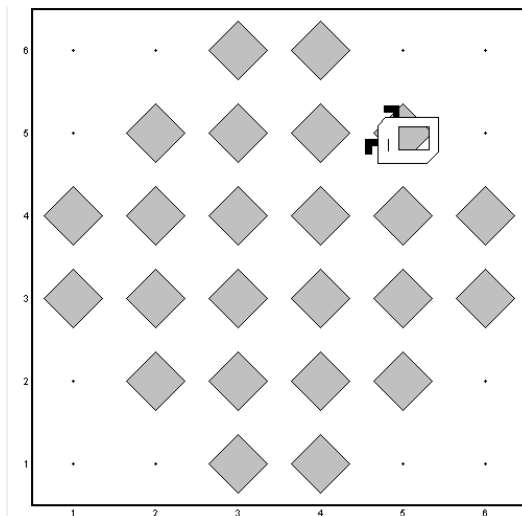
Case #3 Odd rows Even columns:



Case #4 Even rows Odd columns:



Case #5 When rows or columns are less than 7:



As you can see there are no inner square chambers so in this case its not possible to divide into 4 + 4.

After finding what the optimal division is, the problem becomes how to walk on that division while ensuring to make the chambers with the lowest number of moves and i will explain it in the next section.

## 3. Solution

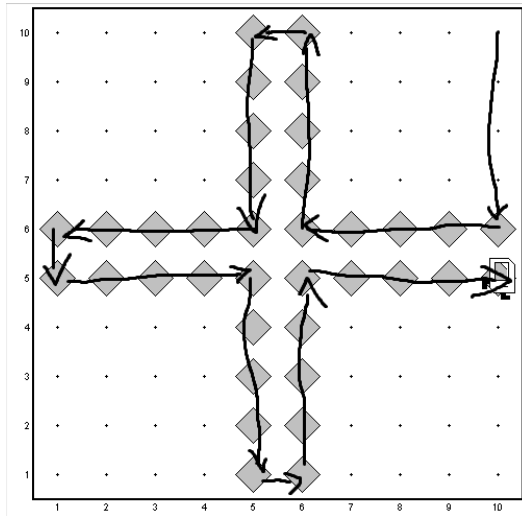
### 3.1 Calculating the Dimensions

To begin with, the initial step entails calculating the dimensions of the map in order to determine the appropriate approach and assess whether the map is of sufficient size for Karel to operate effectively. To accomplish this, I implemented a function called `countDimensions()`, This methodically designed function facilitates the systematic counting of rows and columns within the map. The process involves instructing Karel to traverse eastward until encountering a wall, subsequently proceeding northward while continuing the movement until another wall is encountered. During this traversal, a dedicated counter precisely calculates the number of columns and rows accordingly.

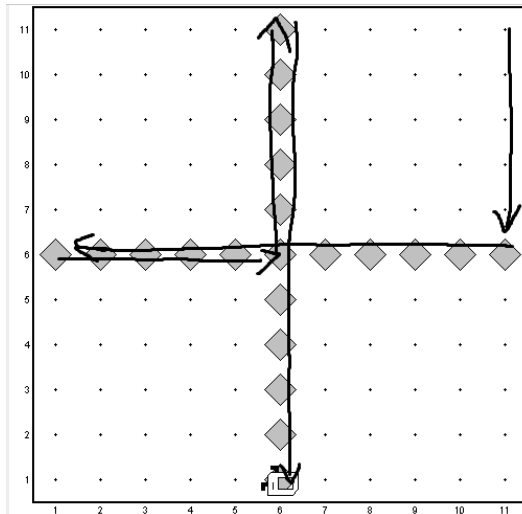
## 3.2 Drawing the Outer Chambers

Following the preliminary assessment of Karel's feasibility in the given map, the subsequent step involves the delineation of the outer four chambers through a fair partitioning of the map into four equal spaces. Commencing from the northeastern corner, Karel undertakes this task with precision. Each unique configuration presents an optimal solution, all of which have been methodically included within the function `drawCross()`.

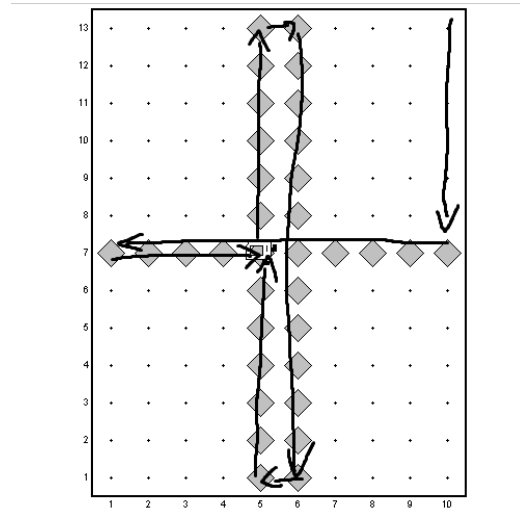
Case #1 Even rows Even columns



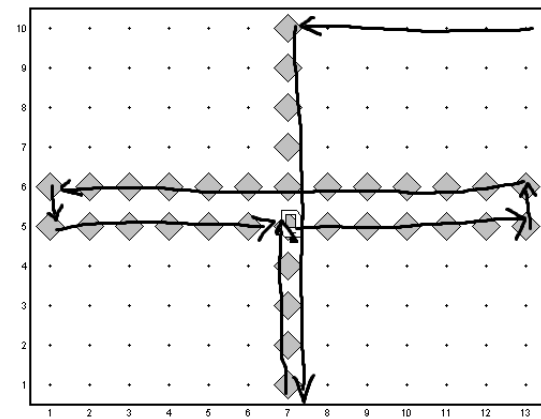
Case #2 Odd rows Odd columns



### Case #3 Odd rows Even columns



### Case #4 Even rows Odd columns

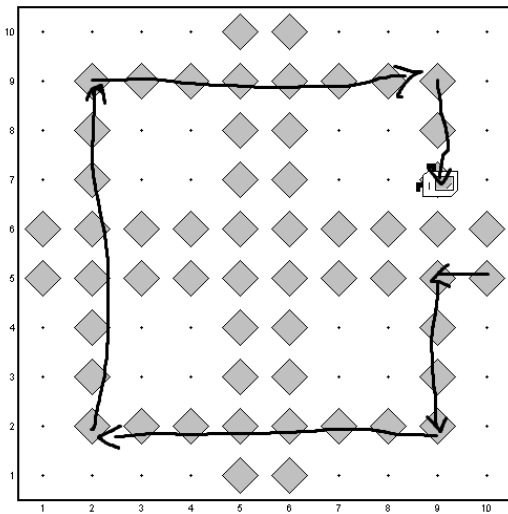


Through this approach, Karel adeptly determines and executes the most optimal path to methodically construct the outer chambers. By employing a calculated plan, Karel skillfully navigates the map, ensuring the creation of the outer chambers with utmost efficiency and precision.

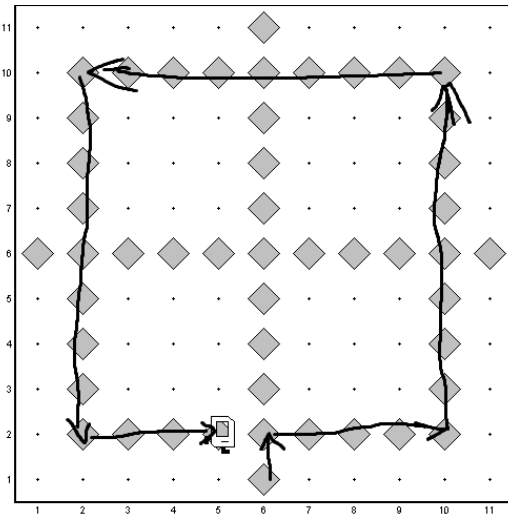
### 3.3 Drawing the Inner Chambers

Following the preceding stages of the map division process, the subsequent and final step entails the creation of the inner chambers. Commencing from Karel's last known position, a methodical and optimal approach is employed to establish the inner chambers. At each subsequent position encountered, Karel thoughtfully analyzed the most efficient course of action to construct the inner chambers. Notably, every unique case is painstakingly implemented and thoroughly addressed within the methodically designed function `drawSquare()`.

Case #1 Even rows Even columns

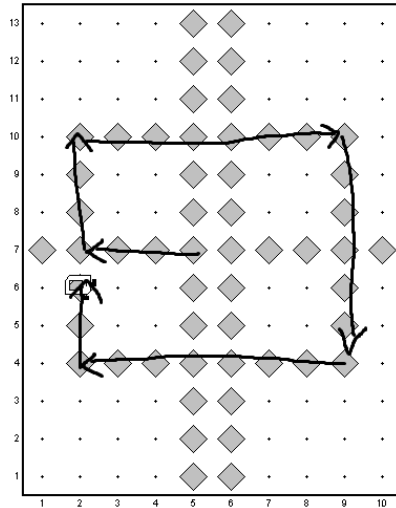


Case #2 Odd rows Odd columns

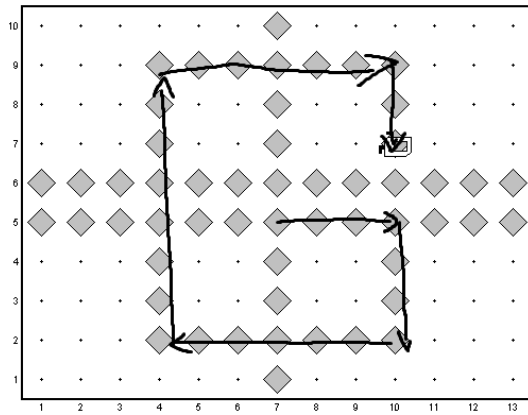




### Case #3 Odd rows Even columns



### Case #4 Even rows Odd columns



The methodical implementation of the function ensures that the previously mentioned approach represents the most optimal methodology for attaining the most spacious inner chambers while minimizing the number of moves required. Consequently, this method facilitates the seamless completion of the map division process with utmost efficiency.

## 4. Overall Algorithm

As a conclusion, the algorithm general steps are as follows:

1. Find the size of the rows and columns.
2. Validate if it's possible for Karel to work or not.

3. Start Dividing the outer chambers based on the cases.
  - a. Even rows Even columns: will move to the middle of the rows and divide the map as the pattern shown above.
  - b. Odd rows Odd columns: will move to the middle of the rows and divide the map in a linear way but it's the most optimal just like the pattern shown above.
  - c. Odd rows Even columns: will move to the middle of the rows and divide the columns in a linear way but the rows in a different way like the pattern above.
  - d. Even rows Odd columns: will move to the middle of the columns and divide the rows in a linear way but the columns in a different way like the pattern above.
4. Then start dividing the inner chambers based on the cases as well.
  - a. Even rows Even columns: according to a simple formula he will move based on whichever is smaller rows or columns, he will move  $(\text{halfc}-\text{height}/2)$  which will determine how much he needs to move from the easternmost to the start of the inner chamber, then  $(\text{height}-1)$  he will move like this all around till he gets the inner chamber.  $\text{height}$  being the smaller between the columns and rows -1, and  $\text{halfc}$  being the half of the columns.
  - b. Odd rows Odd columns: according to a simple formula he will move based on whichever is smaller rows or columns, he will move  $(\text{halfr}-\text{height}/2)$  which will determine how much he needs to move from the southernmost to the start of the inner chamber, then  $(\text{height}-1)$  he will move like this all around till he gets the inner chamber.  $\text{height}$  being the smaller between the columns and rows -1, and  $\text{halfr}$  being the half of the rows.
  - c. Odd rows Even columns: according to a simple formula he will move based on whichever is smaller rows or columns, he will move  $(\text{height})$  which will determine how much he needs to move from the middle of the map to the start of the inner chamber, then  $(\text{height})$  to the north, then  $(\text{height}*2+1)$  to the east, then  $(\text{height}*2)$  to the south, then  $(\text{height}*2)$  to the west and finishing that with  $(\text{height}-1)$  to the north.  $\text{height}$  being (the smaller between the columns and rows -1)/2.

- d. Even rows Odd columns: according to a simple formula he will move based on whichever is smaller rows or columns, he will move  $(height)$  which will determine how much he needs to move from the middle of the map to the start of the inner chamber, then  $(height)$  to the south, then  $(height*2)$  to the west, then  $(height*2+1)$  to the north, then  $(height*2)$  to the east and finishing that with  $(height-1)$  to the south.  $height$  being  $(\text{the smaller between the columns and rows} - 1)/2$ .