



Karel Assignment

Done by

Noor Braik

Hosted by

Dr. Motasem Aldiab

Introduction:

Karel is a very simple robot living in a very simple world. By giving Karel a set of commands, you can direct it to perform certain tasks within its world. In this context we perform dividing a given map into 4 equal chambers or divide them into the maximum possible number of equal chambers (1, 2, or 3, with minimum number of movement and number of beepers.

Links:

GitHub Repository: https://github.com/noor848/Karel_Robot

YouTube link: <https://youtu.be/3xZ6tEbYtgM>

Tools:

- IDE - IntelliJ



- Sublime Merge – Version control tool



- Java 17



Implementation:

The Implementation done by try and error to get the optimal solution

First of all I set number of beepers 1000 as it mentioned in the assignment, then I counted the dimensions of row and columns of each shape as shown below:

```
public void run() {  
    beepers = 0;  
    steps = 0;  
    setBeepersInBag(1000);  
    countRowAndColumn(); // count the row and column  
    System.out.println("Row = " + row);  
    System.out.println("Column = " + column);  
}
```

I divide the shapes according to its dimensions into two cases and each case has its own cases.

Case 1:

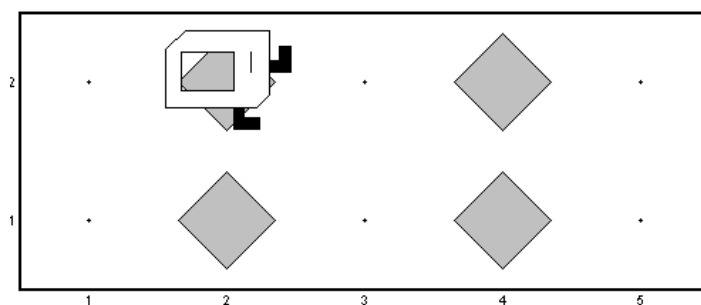
- Row or column less than 4

Has two conditions when

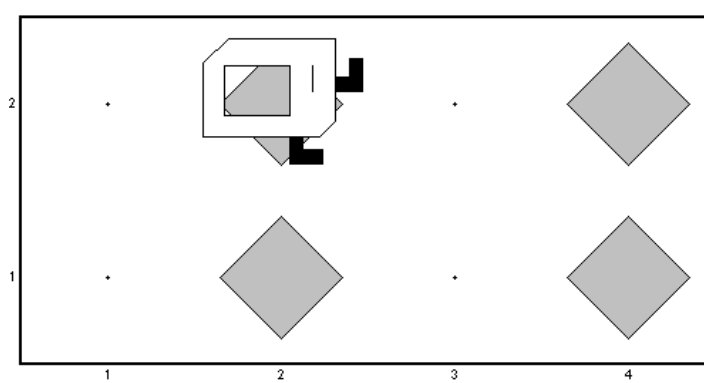
- dimensions' shape less than 7

```
if (column != 0 && row != 0 && (column * row) > 1) {  
    if (row < 4 || column < 4) {  
        if (row <= 7 && column <= 7) {  
            if (column > row && isEven(column)) divideEvenHorizontalDimensionShape(row, column);  
            if (column >= row && isOdd(column)) divideOddHorizontalDimensionShape(row, column);  
            if (column < row && isOdd(row)) divideOddVerticalDimension(column, row);  
            if (column < row && isEven(row)) divideEvenVerticalDimension(row, column);  
        }  
    }  
}
```

For odd we divide the grid with a move step and a step with adding a beeper sequentially



For even we divide the grid with a step adding a beeper and a move step sequentially.



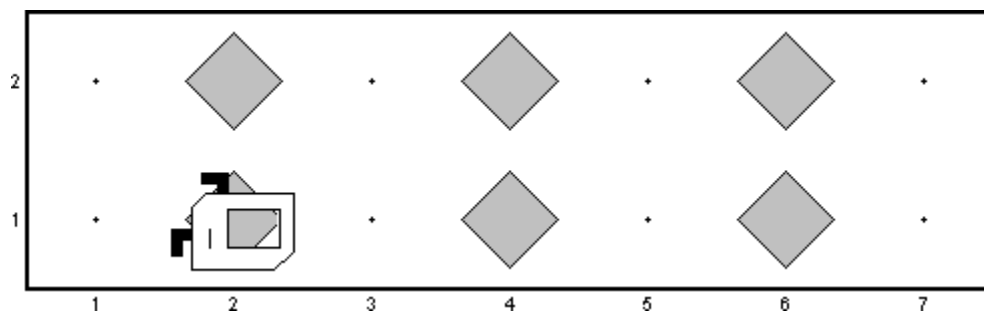
And same things for the vertical shape

- dimensions' shape greater than 7

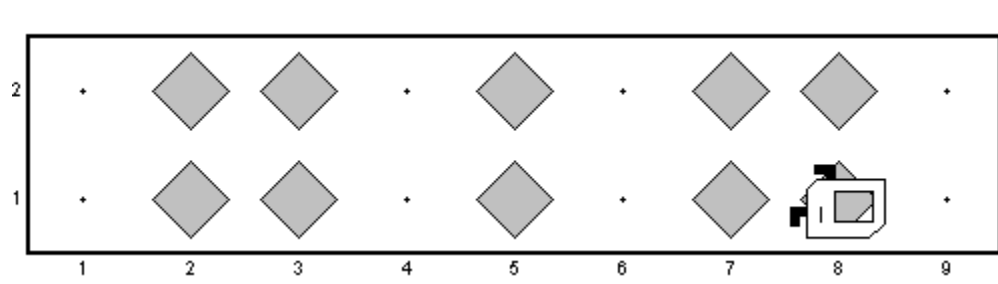
```
else {  
    if (column > row) divideHorizontalDimensionShape(row, column);  
    else divideVerticalDimensionShape(column, row);  
}
```

Note for even parts we need two lines to divide things equally

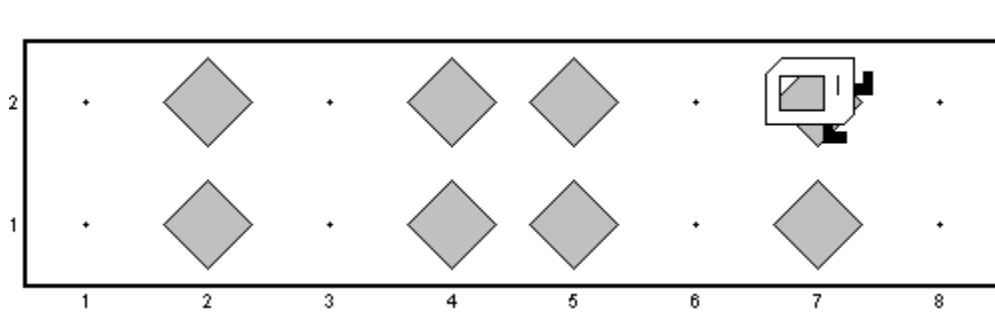
If shape is odd, we divide the main shape into halves and each half check if its odd the division would be as shown below:



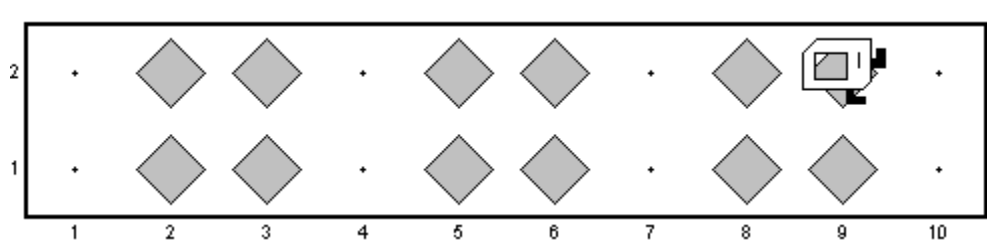
If halves are even the division would be as shown below:



If shape is **even**, we divide the main shape into halves and each half check if its odd the division would be as shown below:



If halves are even the division would be as shown below:



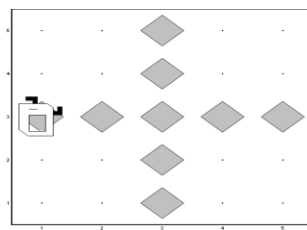
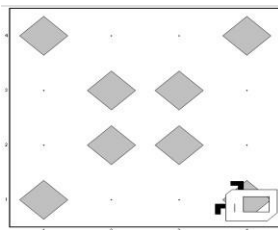
And same things for the vertical shape

Case 2:

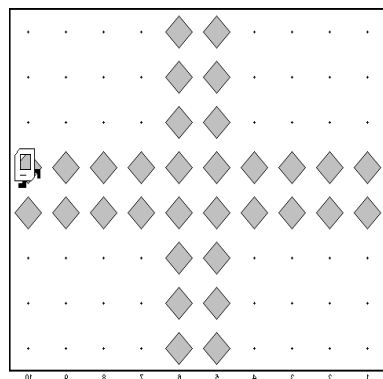
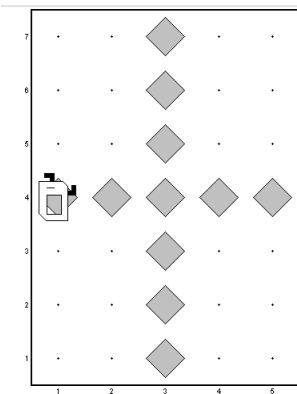
- Row or column greater than 4 or equal

```
else {    /// case row and col >= 4
    if (isEven(column) && isEven(row) && column == row) divideEvenDimensionsSquareShape();
    else if (isEven(column) && isEven(row) && column != row) divideEvenDimensionsRectangleShape(column, row);
    else if (isOdd(column) && isOdd(row)) divideOddDimensionsShape(column, row);
    else {
        if (isOdd(column) && isEven(row)) divideOddColumnEvenRowShape(column, row);
        if (isEven(column) && isOdd(row)) divideEvenColumnOddRowShape(column, row);
    }
}
```

- Even dimensions are equal
- Odd dimensions are equal

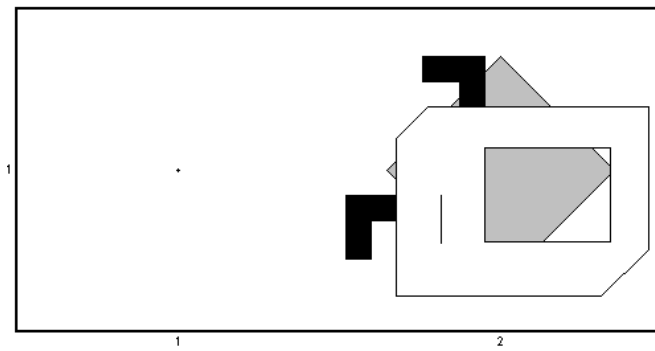


- Odd dimensions not equal
- Even dimensions not equal



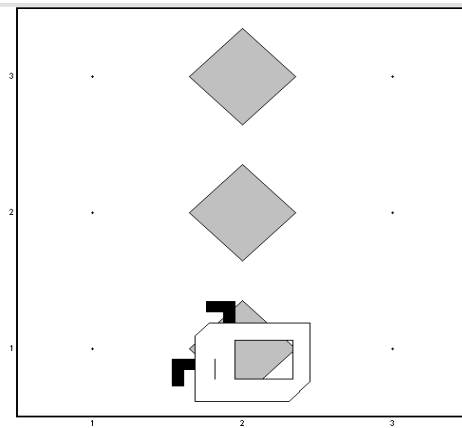
Special cases:

- **One chamber**
1x1, 1x2, 2x2



- **Two Chambers**

columns or row less than 4 and one of them less than 5



- **Three Chambers**

columns or row less than 4 and one of them less than 7

