Karel Assignment.

Train Karel To Get Dimension:

So, first of all, I have to know that Karel doesn't know anything except the functions related to it in the reference card, so he doesn't know about the map dimensions, so first of all I need to train Karel to know the dimensions.

So our Karel always starts from the Bottom Left, so instead of making Karel go through the entire map to check the dimensions, I made it go from the Bottom Left to the Bottom Right using while loop and move() function and now Karel knows the first dimension which is the number of column (X-Axis), and made it go from the Bottom Right to the Upper Right after rotating his direction to move above and used loop and move() function also, So now when Karel Arrived to the Upper Right point it also knew about the other dimension which is the number of rows (Y-Axis).

Dividing Map Into 4 Equals Champers:

Ok, after teaching Karel about dimension, now Karel need to know how to divide the map into 4 equals champers, first thing came to my mind is to think about equality of dimensions such that 3x3 in this case you can divide it whatever you want diagonally, horizontally, and vertically but my Karel is in the Upper Right point.

So dividing the map diagonally will give an advantage across the other techniques, because this step will minimize the number of steps that Karel has to move and also the number of Beepers if we talk about even equality. (Optimization point)

So ok, now let's consider our cases in both Axes bigger than or qual to 3:

1- Odd # of rows & Odd # of columns.
2- Odd # of rows & Even # of columns.
3- Even # of rows & Odd # of columns.
4- Even # of rows & Even # of columns.

So, let’s assume having the **first case**:

In this case the number of rows and columns are odd, dividing the map is done through start dividing vertically from the number of columns / 2 and start putting beepers across the column and when done move Karel to the beginning of the row that is came from numbers of rows / 2 and start fill the beepers across the row.

In this case the beepers would intersect in the middle point of the map so I skipped putting a beepers again in this point responding to the criteria which is minimizing the number of beepers as much as possible. (Optimization point)

Now the **second case** which is odd rows and even columns, in this step we can achieve 4 equals champers by moving Karel to the point number of columns / 2 – 1 and start putting the Beepers across the column and moving to the following column and do the same, then move Karel to the row in which # of rows / 2 and start putting the Beepers across the row and avoid adding multiple Beepers when intersection happened.

In this case, in order to achieve the 4 equals chambers the double line of beepers are required vertically.

So now, let’s move to the **third case** which is having even number of rows and odd number of columns, in this case you will need to add double line of Beepers also, but this time you will add horizontally from the point in which # of rows / 2 – 1 and start putting Beepers then move to the following row and do the same after filling the vertical line of Beepers.

Now in the **4th case** I have even numbers of rows and even numbers of columns.

Ok, this case force me to use double line of Beepers twice, once with the vertical dividing and once with the horizontal, so I moved my Karel to the point in which # of columns / 2 – 1 and filling it with Beepers, then moving to the next column and do the same, after that move Karel to the point in which # of rows / 2 – 1 and fill it with Beepers and do the same with upper row.

Ok, now these all the cases that come to mind if the both axes are bigger than 3, **now what about having one or 2 columns and 7 or more rows, or having one or 2 rows and 7 or more columns**, in this case we also can split the map into 4 equals chambers, but you should keep in mind oddness and evenness, so let's consider having 1 or 2 rows and 7 or more columns, in this case first we need to check if the number of columns is divisible by 2 or not, so if not then we have an odd number so we gonna split the middle column by filling it with a beepers, then I saved the number of columns / 2 in a new integer, the idea of this is to check if the number of columns on the both sides left and right are even or odd, to know using single line of beepers or to use double line of beepers, so if it was odd then you need single line of beepers to split each side and if it's even you need double line of beepers for each side, so let's say it was 7 then,

7 / 2 == 3, then move Karel to the point which is the new integer which is 3 divided on 2 all plus 1 and split there then move Karel to the point (newInteger / 2 + 1) * 2 and fill the column also.

And if the number of columns was even then you need to split the middle two columns (using double line of beepers) and check for the rest number of the columns if it even or odd, and split with single line of beepers if odd and double line of beepers if even.

Ok, for vice versa (+7 rows and 1 or 2 columns) it is the same you just need to teach Karel to determine his move according to the number of rows instead of columns.

Ok, now let us see about the other cases such that 1x1, 1x2, 2x1, 1x3, 2x3, and so on, I mean the cases that are less than 3 in one axes and less than 7 in the other axes,

So this cases I handle them manually, so if I have 1x1 you can't split, if you have 1x2 or 2x1 you can't split too, i.e: if we put some beeper there is still one beeper empty(I don't know if this case consider as a splitting into 1 chamber or not).

If we have 1x3 or 3x1 it is gonna be split into 2 equals chambers, the same as 2x3 or 3x2

Now let's consider having 4x1 or 4x2 or 1x4, these cases also gonna be divided into 2 equals chambers using double line of beepers in the middle columns or rows according to on which axis do we have the number 4 as a dimension.

In 5x1 or 1x5 or 5x2 and 2x5, I split it into 3 equals chambers using single line of beepers and for the 6x1 I divided it also into 3 equals chambers using double line of beepers once and single line of beepers as I'll show in my demo.