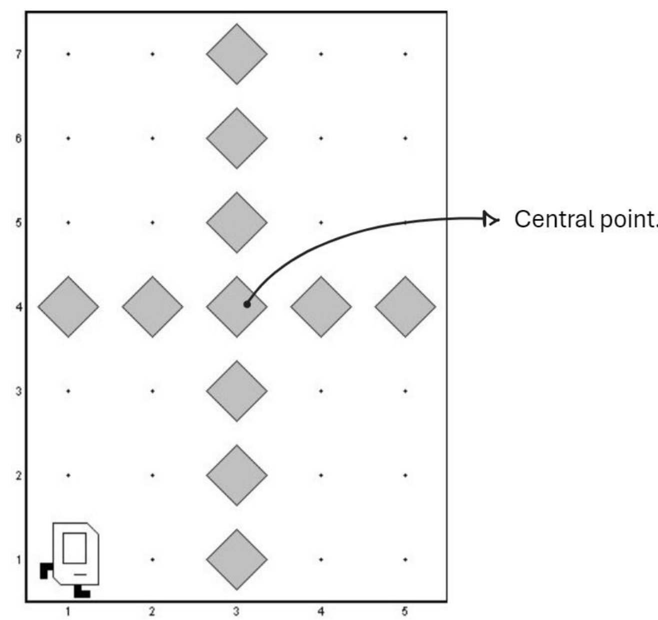
Karel Assignment

First, let us think of each case independently.

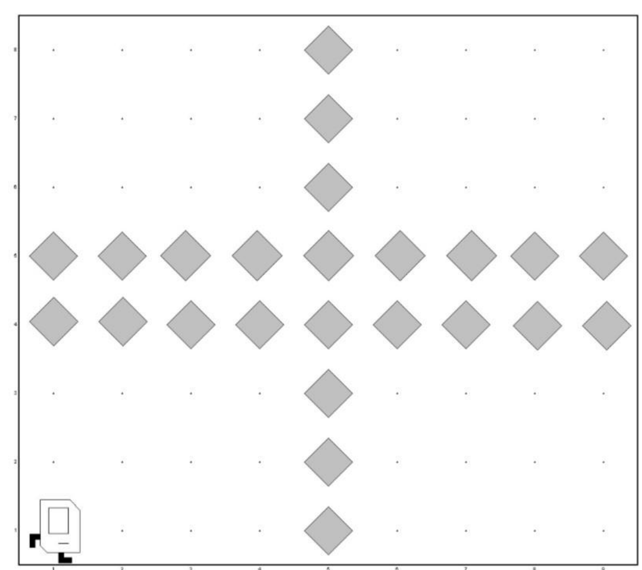
1) Odd x Odd

If the map's dimensions are (Odd x Odd) map, and both are greater than or equal to **3**, then there is a **central point** in the middle of the map that is the center of this map, so we can obviously put beepers along the vertical and horizontal lines that intersect at this central point, and that is the **minimum** number of beepers to put so that the map becomes **4-equal chambers** (which maximize the size of each chamber).



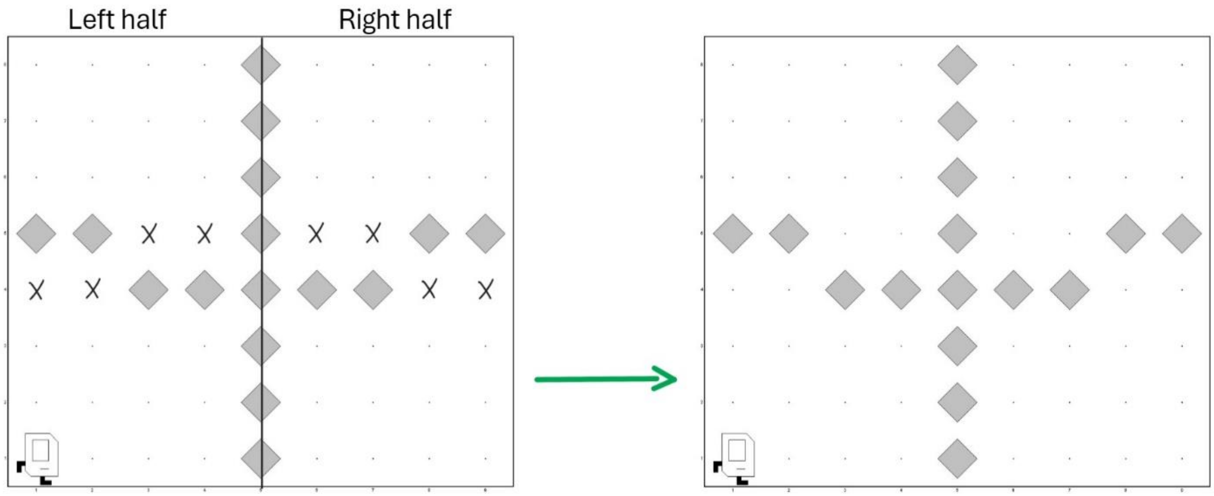
2) Even x Odd

In case we have (Even x Odd) map (both are greater than **2**), we can first split the map into 4-equal chambers by putting beepers in **single** vertical or horizontal line with **double** horizontal or vertical line so that they intersect in the middle of the map, as the follow:



However, **is there any more optimized solution than this**? In other words, can we increase the size of each chamber (which is the same as reducing the number of beepers)?

Yes, we can remove some beepers from the **double line** for each chamber without affecting the equality of the chambers, so if we consider the previous map as two equal halves and for each half at the same time, keep removing beeper by beeper from each chamber oppositely while we can remove (i.e. while the map is still divided into 4 chambers) as the following:

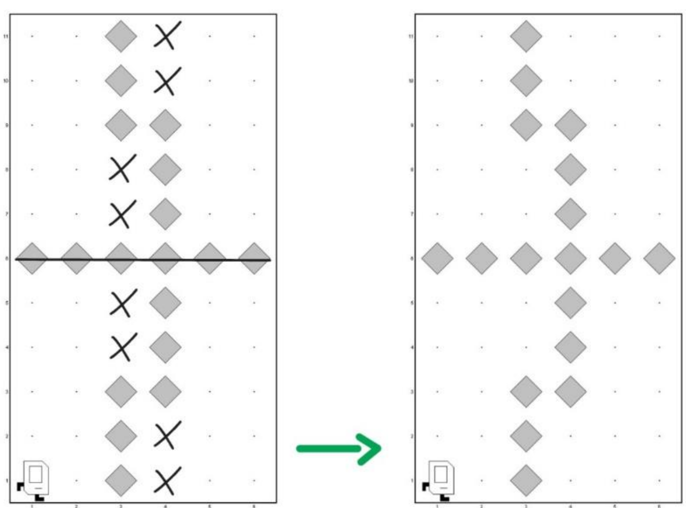


The number of beepers to remove from each chamber is:

$$\textbf{Number of removeable beepers} = \lfloor \textbf{(Odd side length – 1) / 4} \rfloor_{(1)}$$

The size of each chamber will increase by one at a time, until remove all the removeable beepers, and the chambers still have the same size, in this way, we have put as **minimum** beepers as possible to divide the map, which is true because removing **4** more beepers will destroy the (4 chambers) condition.

Another case we may encounter is when the **(Odd side length – 1) / 2 is odd**, applying the formula $_{(1)}$ and removing the beepers will produce the following map, and we are no longer able to remove more beepers.



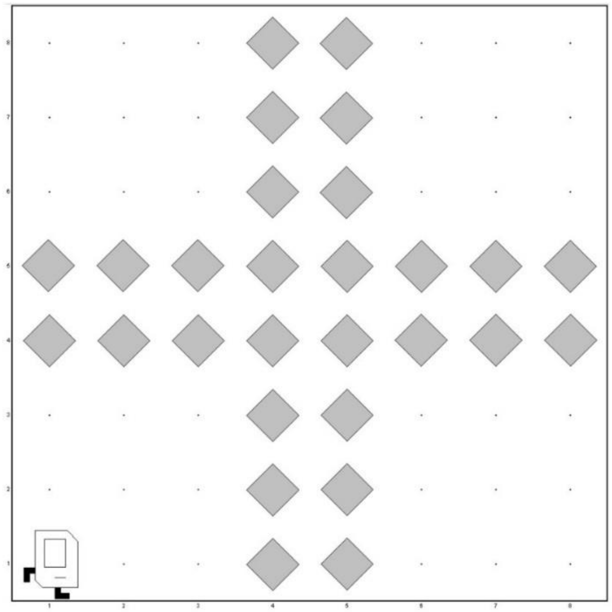
In case we have (**4 x 3** or **3 x 4**), applying the formula $_{(1)}$, we will get **0**, and the size of each chamber will be **1**.

If (**N > 2** and **M > 2**), the following formula tells the **maximum possible chamber size** we could get:

$$= \lfloor \textbf{((N x M) – (N + M - 2)) / 4} \rfloor_{(2)}, \text{ “where } \textbf{N}\text{: Length, } \textbf{M}\text{: Width”}$$

3) Even x Even

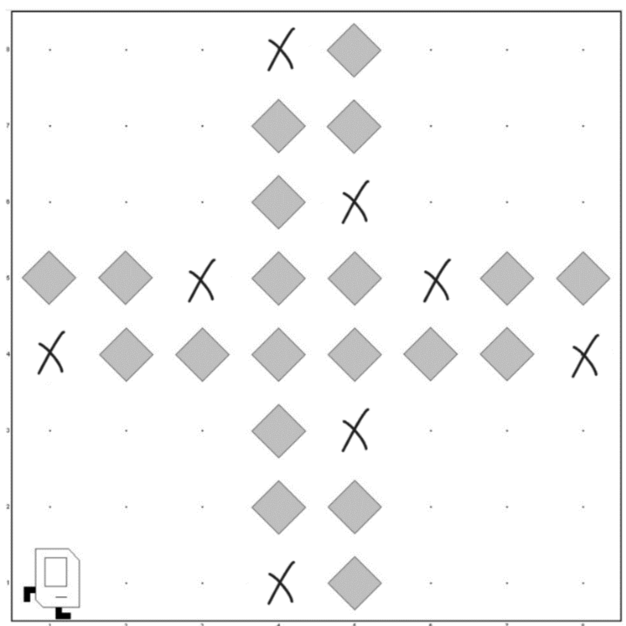
In case of we have (Even x Even) map (and both are greater than **2**), we can divide the map into 4-equal chambers by putting beepers in **double** vertical line with **double** horizontal line, so for the following in **8 x 8** map:



But we know that this is **not the optimal** way to divide the map because we can remove more beepers from both double lines.

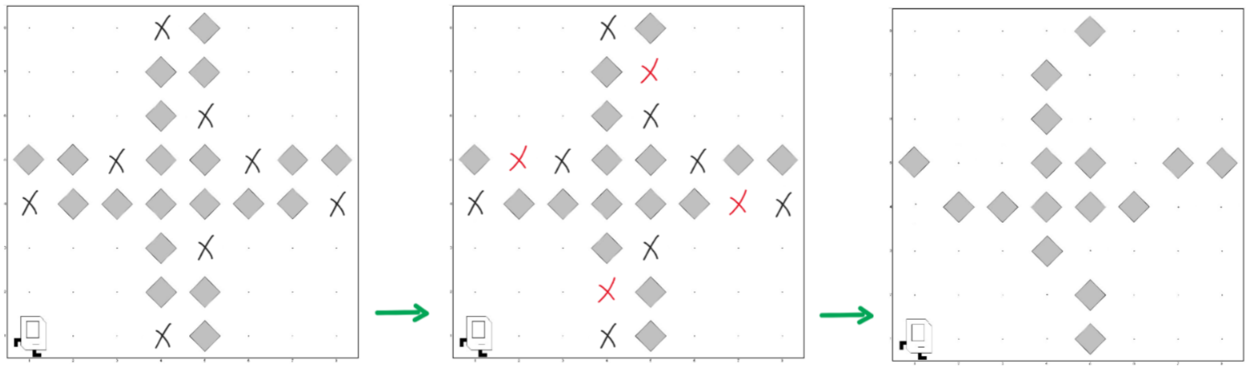
For the following map, the size of each chamber is **9**, applying the formula (2), we will get **12** as the maximum chamber size, so how can we get this chamber size?

If we follow the previous method to maximize the size of the chambers, we will get a chamber size of **11**, as we have removed as maximum beepers as possible from both double lines so that we still have 4 chambers, and they are equal.

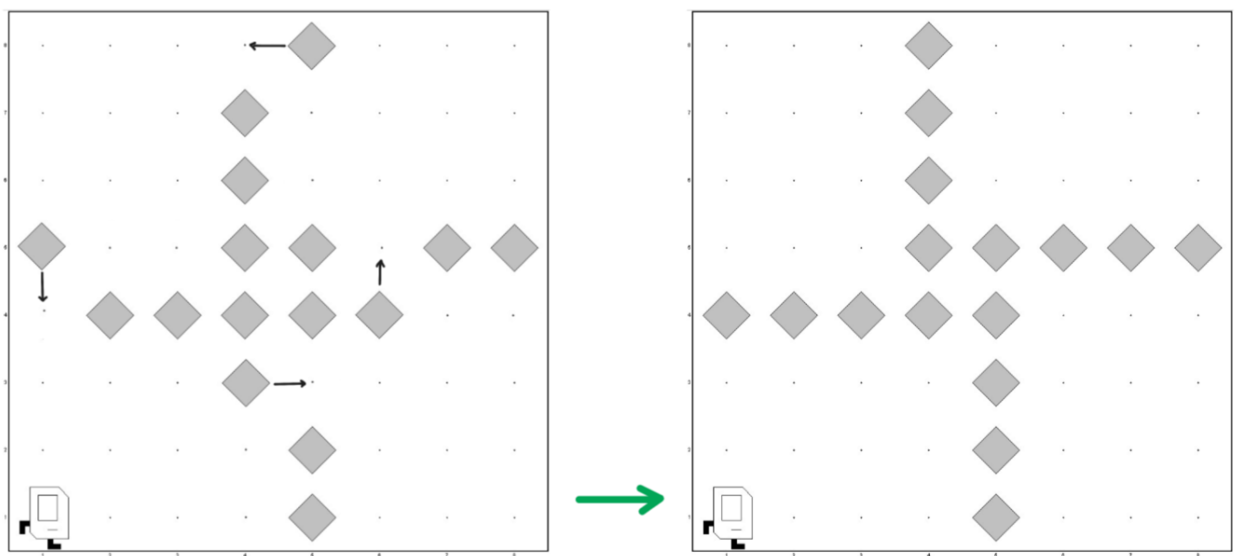


But still**, we can remove** beepers to reach the chamber size of **12**, but how can we do that?

Let's **keep remove beeper** by beeper from each chamber in some pattern without affecting the equality of the chambers:



In this way, we have got **12** chamber size, which is the **maximum size**, but one thing that we should never miss, is to combine as maximum cases as possible with each other so that we can **observe some pattern** and make more reusable functions that solve many cases, so one thing that I will do is to replace the same beepers in way that they are giving more **consistency**:

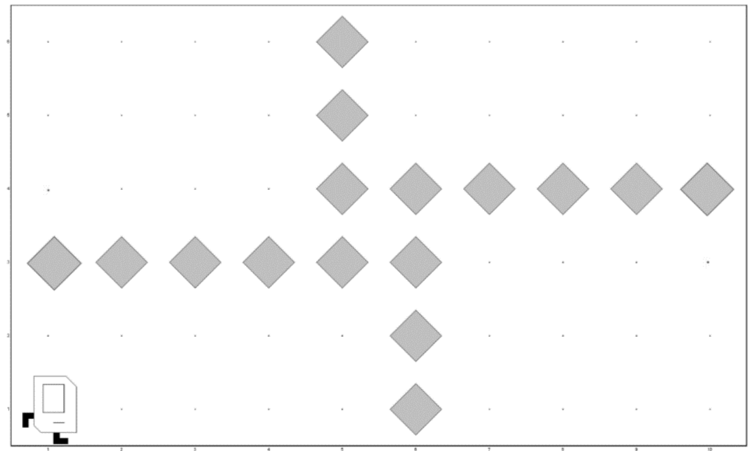


In this way, we have reached the maximum chamber size, and the 4-chambers are still the same size according to the formula (2) (which is **12**).

Next, can we say that if we place the beepers in this way for any (**Even x Even**) map, we will reach the maximum chamber size, and the 4-chambers will have the same size?

Let's see some **other examples** to figure out if we can!

Consider the following (**10 x 6**) map, we will use the previous method to divide this map into 4-equal chambers, and see if it's true to divide it in this way:

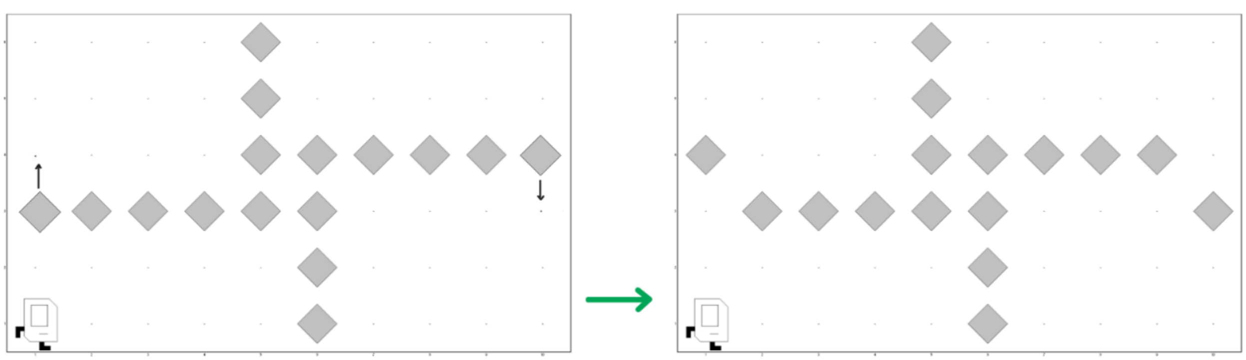


We can see that the chambers **don't have the same size**, so it's **not true** to use the exact method to divide any given Even x Even map, but can we **manipulate** the method such that it still gives the **optimal solution**?

If we look at the size of each chamber, we will see that **2** of them are equal to **12**, and the **other 2** chambers have a size of **10**, so if we move one beeper from each of the smaller chambers and put it in the larger chambers, we will have 4-equal chambers of the size (**11**), which is the maximum size we can get, back to the formula (2), we will see that the maximum size that we can get is:

$$\lfloor ((6 * 10 - (6 + 10 - 2)) / 2) \rfloor = \lfloor 11.5 \rfloor = 11$$

So, the Following changes will be done to the map:



In this way, we've got the maximum chamber size according to formula (2) (which is **11**).

We can calculate that the number of the beepers we will move from the smaller chambers to the larger chambers as follow:

-> x = Length, y = Width

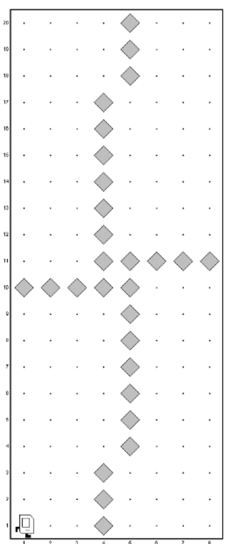
-> $a = (x / 2 - 1) * (y / 2)$ “The size of the (**smaller**/**larger**) chamber”

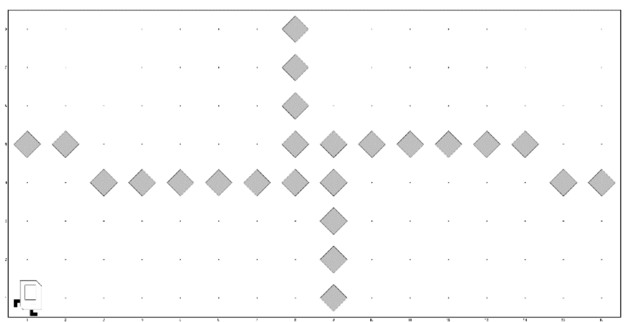
-> $b = (x / 2) * (y / 2 - 1)$ “The size of the (**larger**/**smaller**) chamber”

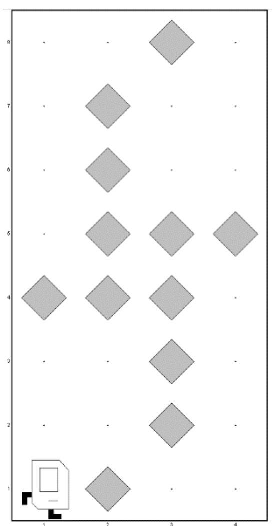
-> Number of beepers to move $= | a - b | / 2$ **(3)**

When we were talking about the (**8 x 8**) map, we haven’t not need to move any beepers to anywhere, and if we apply the formula (3), we will see that the result is **0** because **a** and **b** are equal, but in the previous example, applying formula (3), the result will be **1.**

See the following other examples:







In the first example, the number of beepers to move is:

$$= | (8 / 2 - 1) * (20 / 2) - (8 / 2) * (20 / 2 - 1) | / 2$$

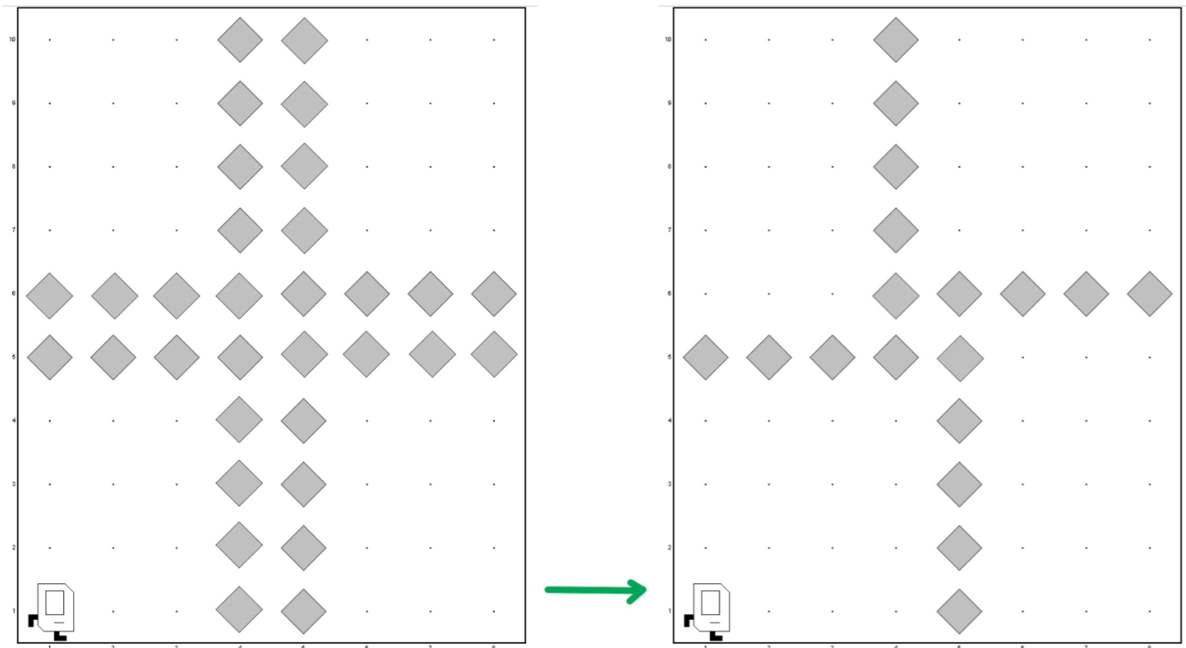
$$= 6 / 2 = 3$$

And for the other two examples, **2** for the second example, and **1** for the third example.

But in all the previous Even x Even cases, the difference between the length and the width is a multiple of **4** (i.e. (**(length / 2) % 2) is equal to ((Width / 2) % 2)**)

(8 – 8 = **0**), (10 – 6 = **4**), (20 – 8 = **12**), (16 – 8 = **8**), (8 – 4 = **4**), let us look at one more example with difference that **is not a multiple** of **4**.

If we look at the following (**8 x 10**), the difference between the length of the map and the width = **2** isn't a multiple of **4** (i.e. (**(8/ 2) % 2 = 0) is not equal to ((10/ 2) % 2 = 1)**), let's see what happens if we follow the same method to place the beepers so that the map is divided to 4-equal chambers and the size of the chamber is maximized.



We can see that the chambers don't have the same size, so can we follow the previous method and apply the formula $_{(3)}$ to calculate the number of beepers to move?

The answer is, **No**.

Applying the formula $_{(3)}$:

$$= |\ (10 / 2 - 1) * (8 / 2) - (10 / 2) * (8 / 2 - 1)\ |\ / 2$$

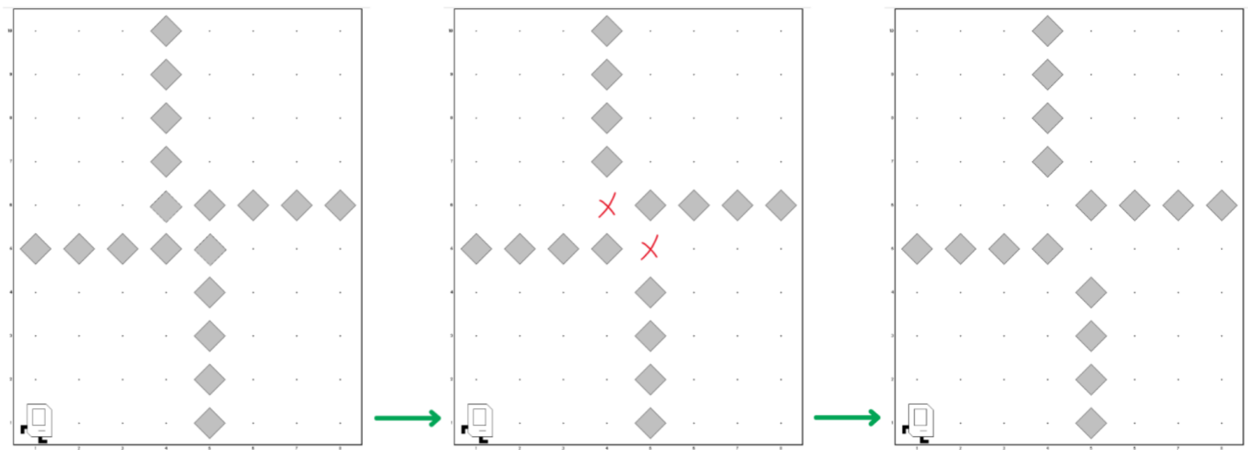
$$|16 - 15| / 2 = 1 / 2 = 0.5$$

The difference between the size of the larger chamber and the smaller chamber is **1** (odd number), which is not divisible by **2**, and we can't move **0.5** beeper from the smaller chamber to the larger chamber so we can make the chambers equal.

According to the formula $_{(3)}$, the value **| a – b |** will always be an odd number if we have an **Even x Even map, And the difference between the length and the width isn't divisible by 4**, so we can't move **| a – b | / 2** beepers because this value will have a fraction of **0.5**.

Let us look at the (**8 x 10**) map one more time and see if we can do anything better than this.

What if we remove instead of moving some beeper?

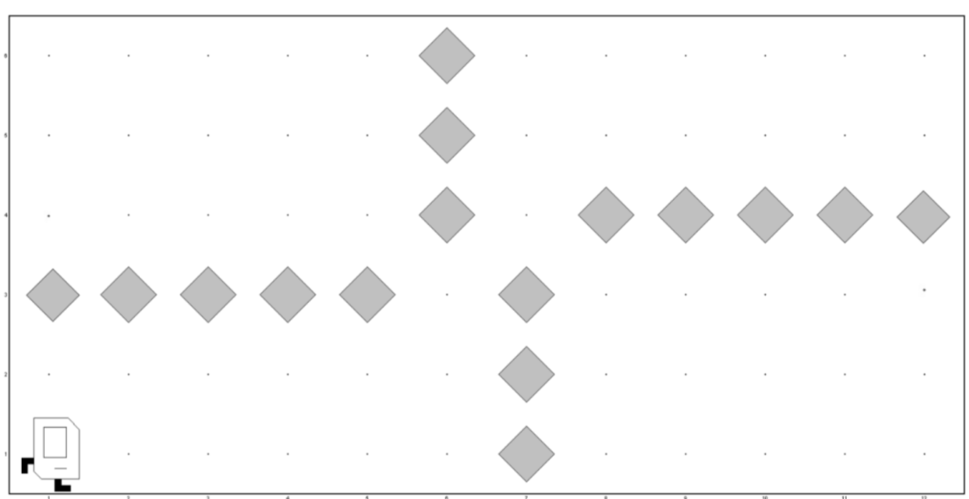


In the above example, we have removed one beeper from each of the smaller chambers, and with this optimization we have reached the maximum possible chamber size according to formula (2), which is **16**, and it is the optimal because removing one more beeper will make the number of chambers equal to **3**, so this is the optimal solution for this case.

Now, is it true to place the beepers in that way for all Even x Even cases that have a dimension difference of a number that is not divisible by **4**?

Let us take another example with (**12 x 6**) map, the dimension difference is **6**, which is not a multiple of **4**.

Following the same way to place the beepers would give as the following chambers:



We can see that the chambers don't have the same size, but the difference between the size of the chambers is **even**!

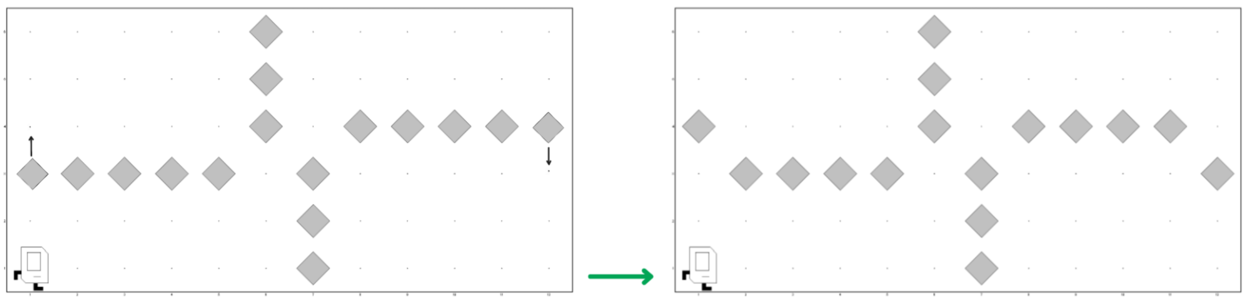
So, we can use the formula $_{(3)}$ to calculate the beepers that we will move them from the smaller chambers to the larger chambers (**but adding one to the minimum chamber in the formula because we have removed one beeper from each of the smaller chambers in our strategy**).

Applying formula $_{(3)}$ with the modification would give us the following:

$$= | (12 / 2 - 1) * (6 / 2) – (12 / 2) * (6 / 2 - 1) | / 2$$

$$|15 – (12 + 1) | / 2$$

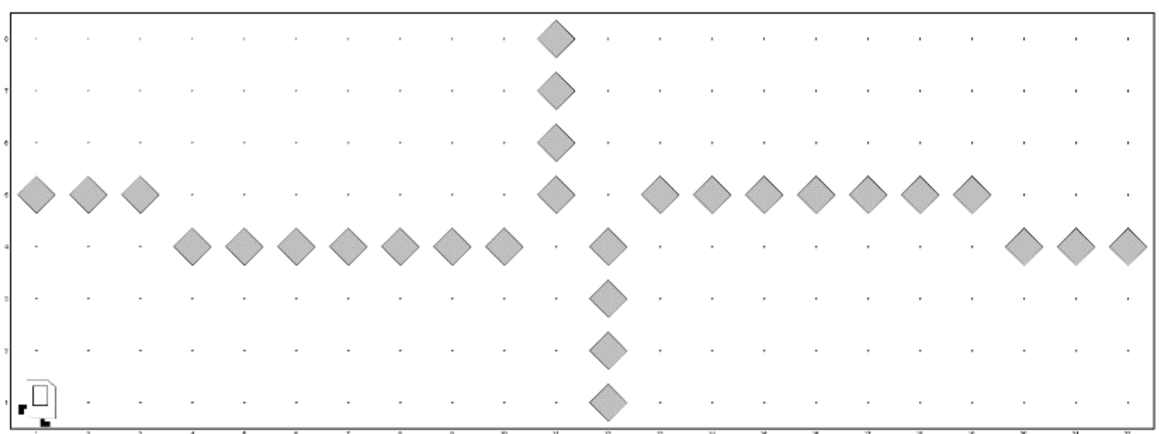
$$= 2 / 2 = 1$$



And this would give us a chamber size of **14**, which is the maximum size we can get according to formula $_{(2)}$.

So, if we have the following (**22 x 8**) map, applying formula $_{(3)}$:

$$= | (22 / 2 - 1) * (8 / 2) – (22 / 2) * (8 / 2 - 1) | / 2 = | 40 – (33 + 1) | / 2 = 6 / 2 = 3$$



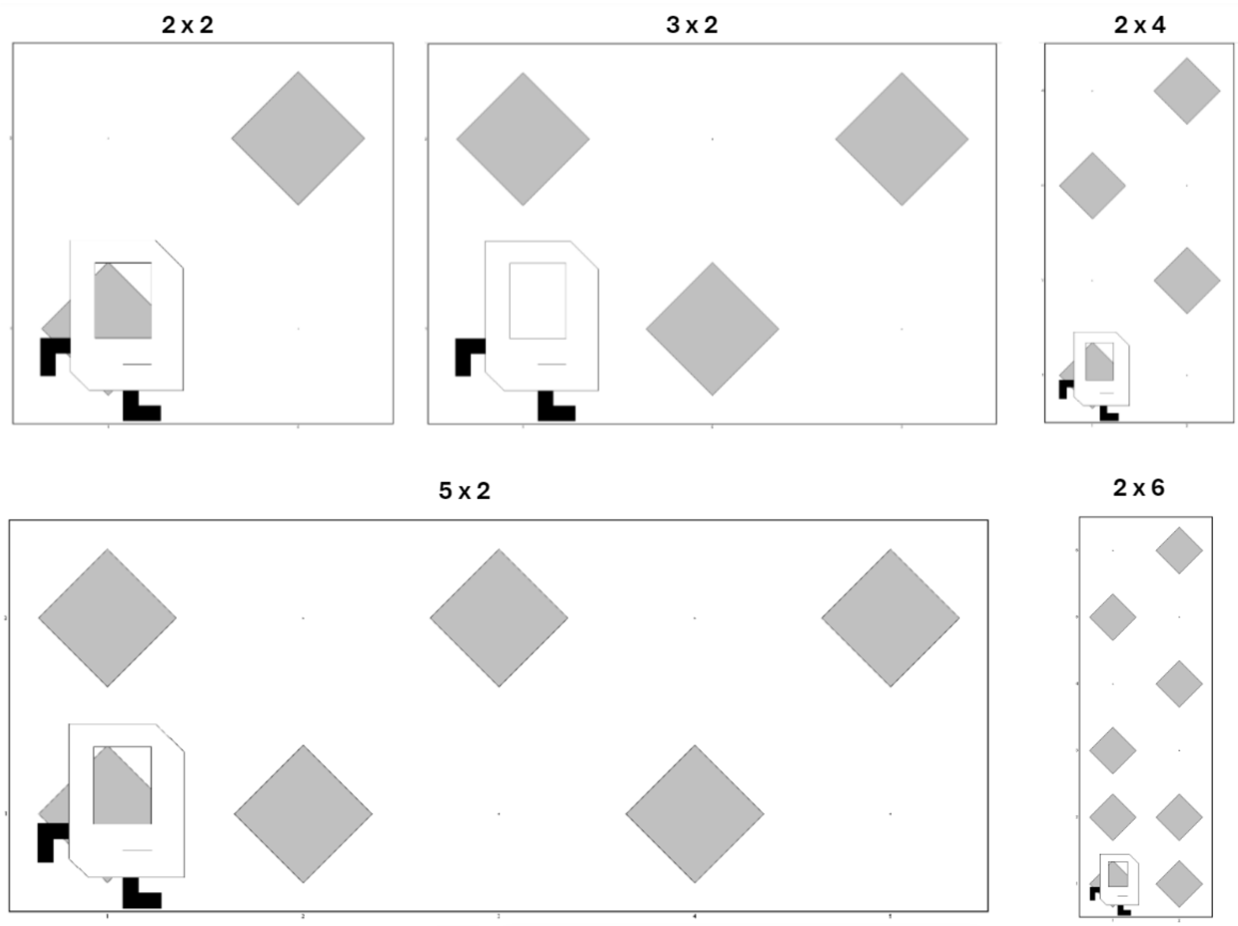
With this strategy, we've got a chamber size of **37**, which is the maximum chamber size according to the formula $_{(2)}$.

4) Special cases

All the above strategies work for dimensions that are greater than **2**, what if we have a length or width that is less than or equal to **2**?

If we have a dimension of (**2 x N**) or (**N x 2**) and **N** is greater than **1**, we will do the following:

If N is less that **7**, we will put the beepers in a **Zigzag** movement until we have 4 chambers, then we will fill the remaining blank cells with beepers, see the following examples:



In the first three cases (**2x2** and **2x3** and **2x4**) the number of the chambers for each of them in order (**2**, **3**, **4**).

And for the last two examples we still have 4-equal chambers with chamber size of **1**, which is the maximum size we can get, and we fill the remaining blank cells with beepers.

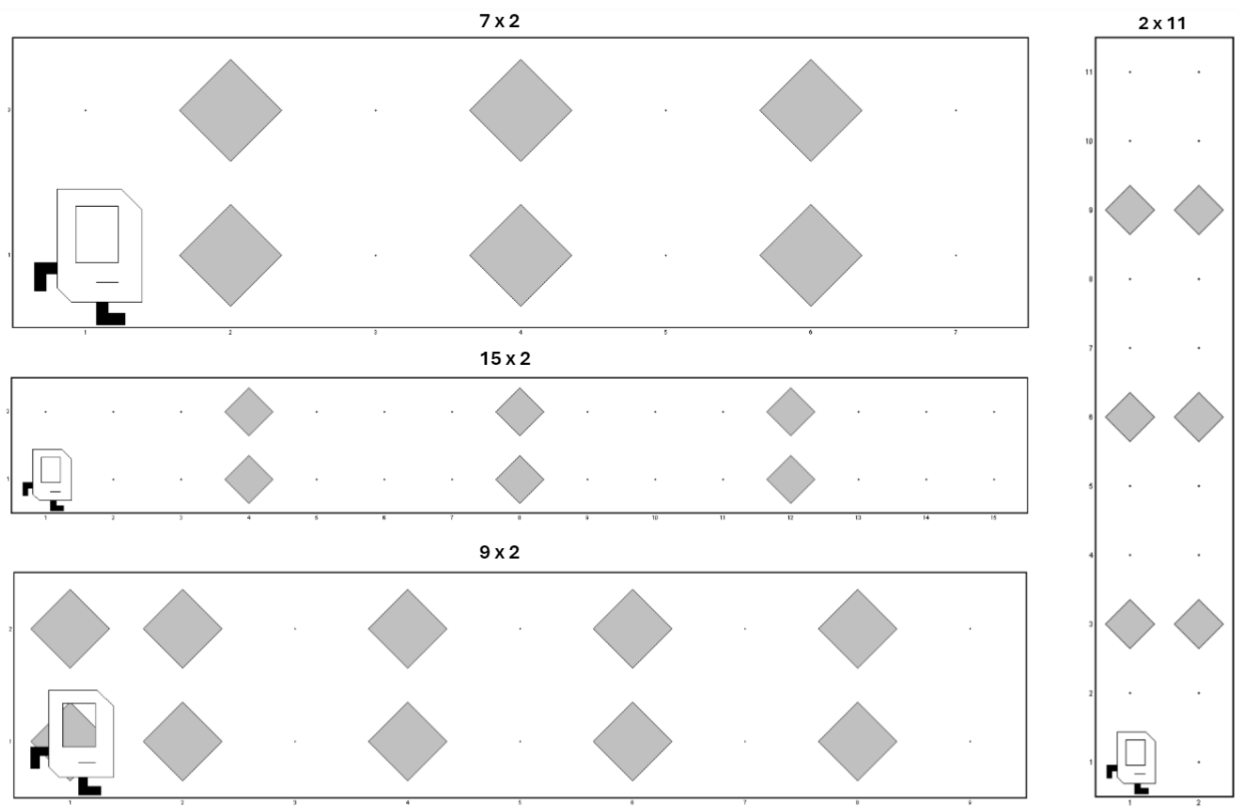
And if we have (**2 x N**) or (**N x 2**), and **N** is greater than or equal to **7**, then the size of each chamber will be calculated as follows:

$$= \lfloor (N - 3) / 4 \rfloor \times 2 \quad (4)$$

Which is optimal because making 4 chambers only requires **3 vertical** (or **horizontal**) lines of beepers between them.

And when we make the 4-equal chambers with maximum size we can get, we will fill the remaining blank cells with beepers.

So, for the following examples:



In the examples that have the dimensions (**7 x 2**) and (**2 x 11**) and (**15 x 2**), we can see that the numbers **7**, **11** and **15** are divisible by **4** if we subtract **3** from each of them, which gives 4 equal chambers with no need to fill any other blank cells.

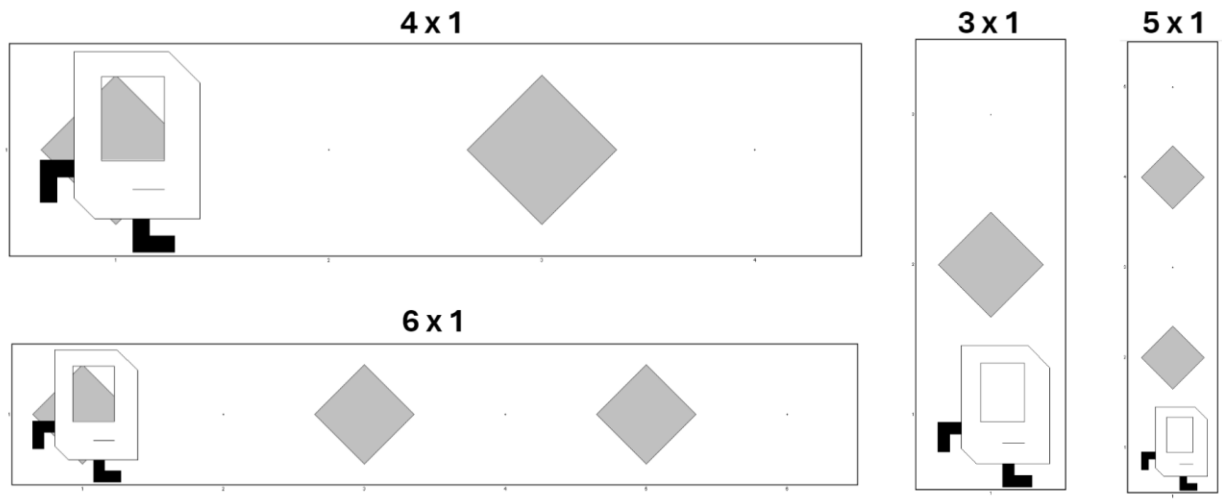
But, For the (**9 x 2**) map, (**(9 - 3) / 4)** is equal to **(1.5)**, and here the fraction indicates that the 4 chambers will not fit at the map, so we need to fill **4** blank cells with beepers in this case.

Notice starting from **N = 7**, which has a chamber size of **2**, we can observe that adding **4** to **N** will increase the chamber size by **2** (i.e. the new chamber size = **chamber size of N + 2**).

Another case is that when we have (**1 x N**) or (**N x 1**), and **N** is greater than **2**, we will keep putting beepers in the following way (until we have **N** that is greater than or equal to **7**):

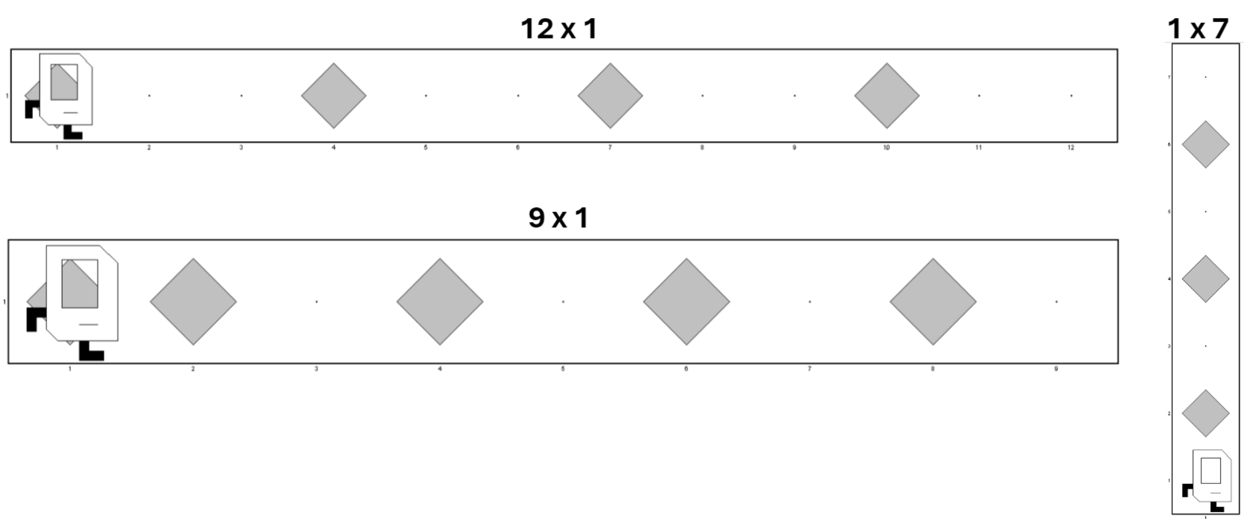
Starting from an end, leave one blank cell and put a beeper on the next cell until reaching the other end.

Let us look at the following examples:



In these cases, we are just trying to make as maximum chambers as possible of the size **1.**

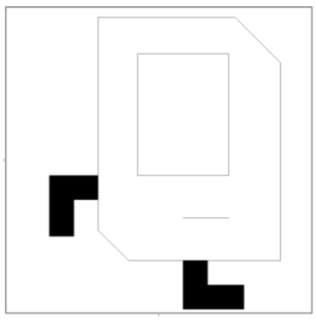
If **N** is greater than or equal to **7**, we will use $\lfloor \mathbf{(N - 3) / 4} \rfloor$ (the formula $_{(4)}$ but without multiplying by **2**) to calculate the size of each chamber and after making all the chambers with the maximum size we can get, we will also fill the remaining blank cells with beepers, so for the following examples:



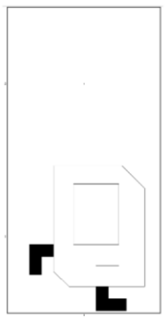
I.e. we will put beepers between the chambers (with the maximum chamber size), and we need one beeper between two chambers, and fill the remaining blank cells with beepers.

Finally, the Last two possible maps that we could face are (**2 x 1** or **1 x 2**) map and (**1 x 1**) map, in these two cases, we will leave the map as it is, because the maximum number of chambers that we could get for each of those maps is **1**, and putting beepers will just minimize the size of this chamber, which is something we don't want to do.

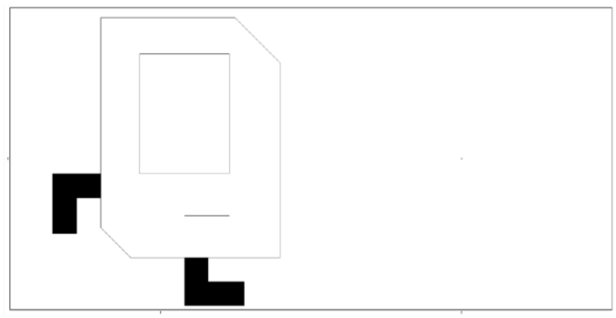
1 x 1



1 x 2



2 x 1



That's all.

Note: in the code, I haven't used "pickBeeper()" function at all, instead, I have just used the previous formulas to make Karel put beeper in some cell iff this cell has a beeper in the result map (the map after dividing it to 4-equal chambers).

Thanks a lot

Ahmad Nabeel Al-Jaber