**Problem 3 - Sunlight**

Sandy is a little girl who spends her free time playing with her friends. Unfortunately, Sandy broke her wristwatch a week ago. Now she is trying a new trick using the sunlight to guess the time. But this trick is useful only when the sky isn’t cloudy. Your task is to help Sandy by writing a program which shows how bright the sun is at the moment.

You are given an integer number **N** (always **odd**), corresponding to the **width and height** of the sun and the **length** of the **horizontal and vertical** sunbeams. The **diagonal** sunbeams have length equal to **N – 1.**

### Input

The input data should be read from the console.

* On the only input line you will be given an integer **N** - the **size** of the sun.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output should be printed on the console. Use the “**\***” (asterisk) to mark the sun and the sunbeams and “**.**” (dot) for the rest. Follow the examples below.

### Constraints

* **N** will always be a positive **odd** number in the range [**3** … **33].**
* Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

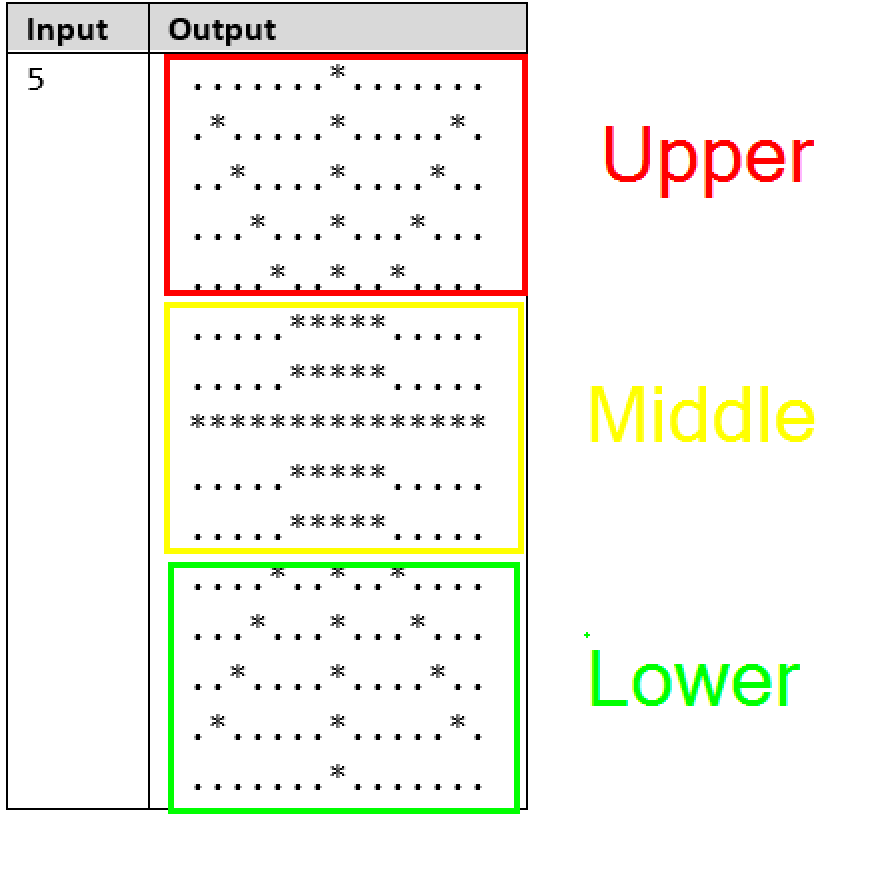
### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 3 | ....\*....  .\*..\*..\*.  ..\*.\*.\*..  ...\*\*\*...  \*\*\*\*\*\*\*\*\*  ...\*\*\*...  ..\*.\*.\*..  .\*..\*..\*.  ....\*.... |  | 5 | .......\*.......  .\*.....\*.....\*.  ..\*....\*....\*..  ...\*...\*...\*...  ....\*..\*..\*....  .....\*\*\*\*\*.....  .....\*\*\*\*\*.....  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .....\*\*\*\*\*.....  .....\*\*\*\*\*.....  ....\*..\*..\*....  ...\*...\*...\*...  ..\*....\*....\*..  .\*.....\*.....\*.  .......\*....... |

**Tasks**

### Step 1: Divide the problem into smaller parts

Read the Problem description and get familiar with the problem. After that, try dividing the logic of creating it into smaller and simpler parts to complete step by step. Let’s try dividing the whole drawing into three parts: **Upper, Middle and Lower Part**.

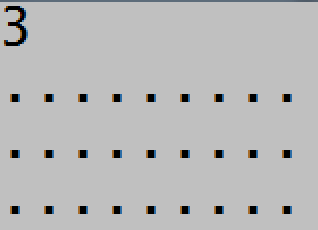


### Step 2: Define some constraints

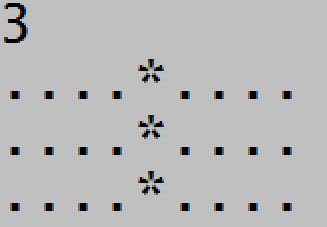
Before beginning to draw, you should define some constraints. First of all, try to determine what is the width and height of the drawing based on the input. Find out how that would be useful and store it in some kind of variables.

### Step 3: Draw the upper part

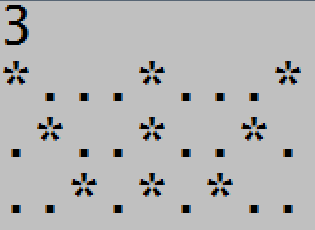
After you have read the input and analyzed the drawing constraints it is time to complete the first part of the task. Firstly, let’s try to draw the field of the upper part of the object. For example:



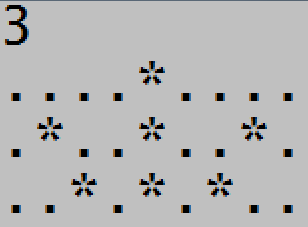
After that, try to determine the starting position of the **middle beam** and how to calculate it according to the input. Next, try to put it in the logic of the graphic and display it on the console.



Now, try displaying the **left and right** beams of the sun. First, determine their position according to the input. After that, figure out how to update their position for each row of the graphic. Try displaying them without any constraints as to the current row.



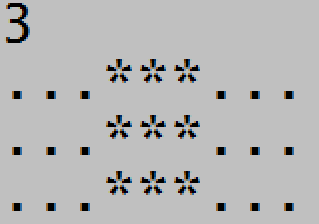
Finally, try determining a constraint for drawing the left and right beam **only if they are not on the first row.**

****

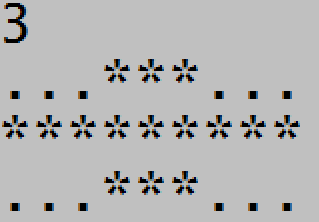
### Step 3: Draw the middle part

After you are done rendering the upper part of the object, let’s try handling the middle part. For a start, try determining the constraints for drawing the **body of the sun –** the starting position of each of it’s fragments, it’s length and so on. Furthermore, don’t forget to determine the width and height of this part of the sun as well.

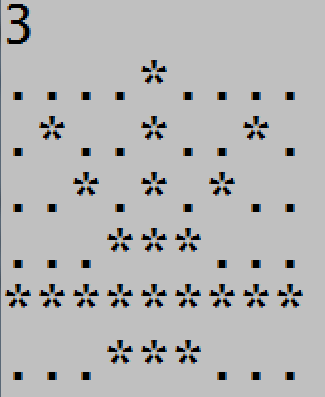
After that, let’s try building up the logic for it’s display. After you have figured out the constraints, try rendering this part of the sun with its body in it.



After that, try figuring out the constraint for drawing the beam going through the middle of the body and handle the logic for displaying it as well.



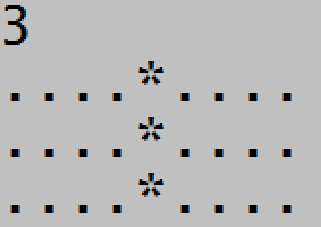
Finally, try combining the rendering of the upper and the middle part of the figure to see our progress. **Also, don’t forget to test your logic with different dimensions as well.**



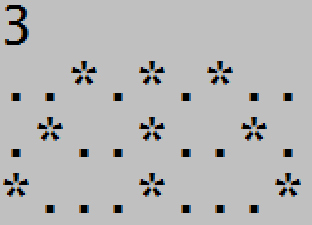
### Step 4: Draw the lower part

We are getting close to our final destination. All that is left is to draw the lower part of our figure. First, let’s try determining what fragments we have here and their constraints. Also, determine the **width and height** of this field as well. Make any initial preparations and variable declarations before starting to draw this part of the figure.

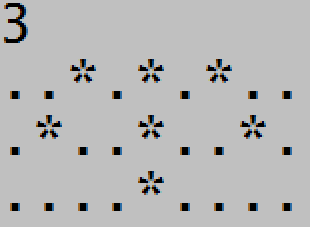
Now, let’s start with the **middle beam.** Determine its position in the graphic and handle the program logic for rendering it.



After that, handle the logic of the **left and right** beams of the sun. Think about what their starting positions are, try calculating them according to the user input and think about how to update them after each row of the graphic. Try displaying them without any constraints at first.



Finally, think about the constraints for the left and right beams, update the program logic and see the results.



### Step 5: Test your solution with different dimensions

We are almost there! Now what’s left is to test your solution with whatever inputs you can think of in order to see whether the graphic is OK according in all cases. **Now you are ready to submit your code.**

