Bonus

Solve the following exercises and upload your solutions to Moodle until the specified due date.

Important Information!

Please try to *exactly match the output* given in the examples (naturally, the input can be different). We are running automated tests to aid in the correction and grading process, and deviations from the specified output lead to a significant organizational overhead, which we cannot handle in the majority of the cases due to the high number of submissions.

Make sure to use the *exact filenames* that are specified for each individual exercise.

Also, use the provided unit tests to check your scripts before submission (see the slides Handing in Assignments on Moodle). Feel free to copy the example text from the assignment sheet, and then change it according to the exercise task to match the output as best as possible.

In this assignment, it is of *particular importance* to wrap the printing that you see in the example outputs in `if __name__=='__main__':`, as your exercises essentially only consist of a function definition. Example - let's say the task is to write a function that doubles the float value that is passed:

```
def double(var: float) -> float:
    return var*2

if __name__ == '__main__':
    print(double(3.4))
```

Unless explicitly stated otherwise, you can assume correct user input and correct arguments.

You are *not allowed* to use any concepts and modules that have not yet been presented in the lecture.

You are allowed in this assignment to implement additional or multiplication attributes and methods as long as the original interface remains unchanged.

Exercise 1 – Submission: `a11_ex1.py` 10 Points

Conway's Game of Life, created by mathematician John Horton Conway in 1970, is a cellular automaton that demonstrates how complex patterns and behaviours can emerge from simple rules. The "game" takes place on a grid, where each cell is either alive or dead, and the state of the grid evolves over discrete steps based on the states of neighbouring cells.

For this assignment, you will implement Conway's Game of Life in Python using modular programming principles. You will divide your implementation into multiple Python modules, each responsible for specific aspects of the program. Each exercise creates one module, with the last exercise providing the entry point of the program.

Grid management

The module for this exercise should provide the basic grid operations of the Game of Life. This includes creating a grid, displaying it, calculating the number of live neighbours for each cell, and calculating the total number of live cells.

The module should contain the following functions:

1. `create_grid(rows: int, cols: int) -> np.ndarray`, which creates and returns a randomly generated 2D array of 0s and 1s. Use `np.random.seed(0)`
2. `count_neighbours(grid: np.ndarray, row:int, col:int) -> int`, which counts the number of alive neighbours, i.e., cells with value 1, around the cell at index `[row,col]`.
3. `print_grid(grid: np.ndarray)`, which prints `grid` to the console. The grid should be displayed as a table with "*" for live cells with 1s and "." for dead cells with 0s. It should additionally print an empty line after the grid.
4. `number_of_live_cells(grid:np.ndarray) -> int`, which returns the total number of live cells in the grid.

Example program execution:

```
grid = create_grid(5, 5)
print_grid(grid)
print(number_of_live_cells(grid))
print(count_neighbours(grid, 0, 0))
print(count_neighbours(grid, 2, 2))
print(count_neighbours(grid, 4, 4))
print(count_neighbours(grid, 0, 4))
```

Example output:

```
. * * . *
* * * * *
* . . * .
. . . . *
. * * . .
13
3
4
1
2
```

Exercise 2 – Submission: `a11_ex2.py` 10 Points

Game Logic

This module will apply Conway's rules to determine how the grid evolves from one generation to the next. Write a function `apply_rules(grid: np.ndarray)` that generates the next state of the grid according to the following update rules:

1. Any live cell with fewer than two live neighbours dies (underpopulation).
2. Any live cell with two or three live neighbours survives.
3. Any live cell with more than three live neighbours dies (overpopulation).
4. Any dead cell with exactly three live neighbours becomes alive (reproduction).

Ensure you properly import `count_neighbours` from a11_ex1.py for this.

Example program execution (assumes `print_grid` to be imported; you don't need that for the actual function):

```
import numpy as np
grid = [
    [0, 1, 0, 1, 0],
    [0, 0, 1, 0, 0],
    [1, 1, 1, 1, 1],
    [0, 0, 1, 0, 0],
    [0, 1, 0, 1, 0]
]
print_grid(np.array(grid))
new_grid = apply_rules(grid)
print_grid(new_grid)
```

Example output:

```
. * . * .
. . * . .
* * * * *
. . * . .
. * . * .

. . * . .
* . . . *
. . . . .
* . . . *
. . * . .
```

Exercise 3 – Submission: `a11_ex3.py` 10 Points

File Handling

This module should handle saving grids to files and loading them from files. For this, we want to save the grids as strings of 0s and 1s.

The module should contain the following functions:

1. `load_grid_from_file(filename: str) -> np.ndarray`, which loads a saved grid from the relative path `filename`. You can assume that the given path contains a valid file, i.e., a text file describing a 2D matrix of 0s and 1s. Each line represents one row, with 0s and 1s indicating dead and live cells, respectively.
2. `save_grid_to_file(grid: np.ndarray, filename: str)`, which saves a given grid to the path `filename`. You can assume that the user will always use a text file as the output path.

Example program execution:

```
grid = load_grid_from_file("initial_grid.txt")
print_grid(grid)
save_grid_to_file(grid, "copy_initial_grid.txt")
```

Example output (assumes `print_grid` to be imported; you don't need that for the actual function):

```
. . . . . .
. . . . . * * . . . . . .
. . . . . * . . . . . .
. . . . . * . . . . . * . . . . . .
. . . * . . . . . * . . . . . * * . . . . . .
. . * * . . . . . * . . . . . * * . . . . . .
. . . . . * . . . . . * * * . . . . . * . .
. . . . . * * . . . . . * * . . . . . * * . .
. . . . . * * . . . . . .
. . . . . * . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

Exercise 4 – Submission: `a11_ex4.py` 20 Points

Main

The main module serves as the entry into the program. It should import all required functions from the other modules to build the Game of Life. It should only have one function `main`.

The workflow of this function is as follows:

- It initialises a grid and asks the user whether the grid should be randomly created or by loading from a file.
  - If the user inputs random initialisation, the program should ask for the requested number of rows and columns.
  - If the user requests to load from a file, the program should ask for the filename.
- It runs the Game of Life on the grid according to the given rules for 10 generations. After every generation, it should print the current generation number and the number of currently live cells.
- After every 10 generations, the program should ask whether to continue with 10 more generations, whether to save the state of the grid, or whether to quit the program. If the user requests to save the grid, the program should ask for a filename, then save the file, and finally continue the program with 10 generations.
- If desired, you can use `time.sleep(0.1)` to slow down the outputs.

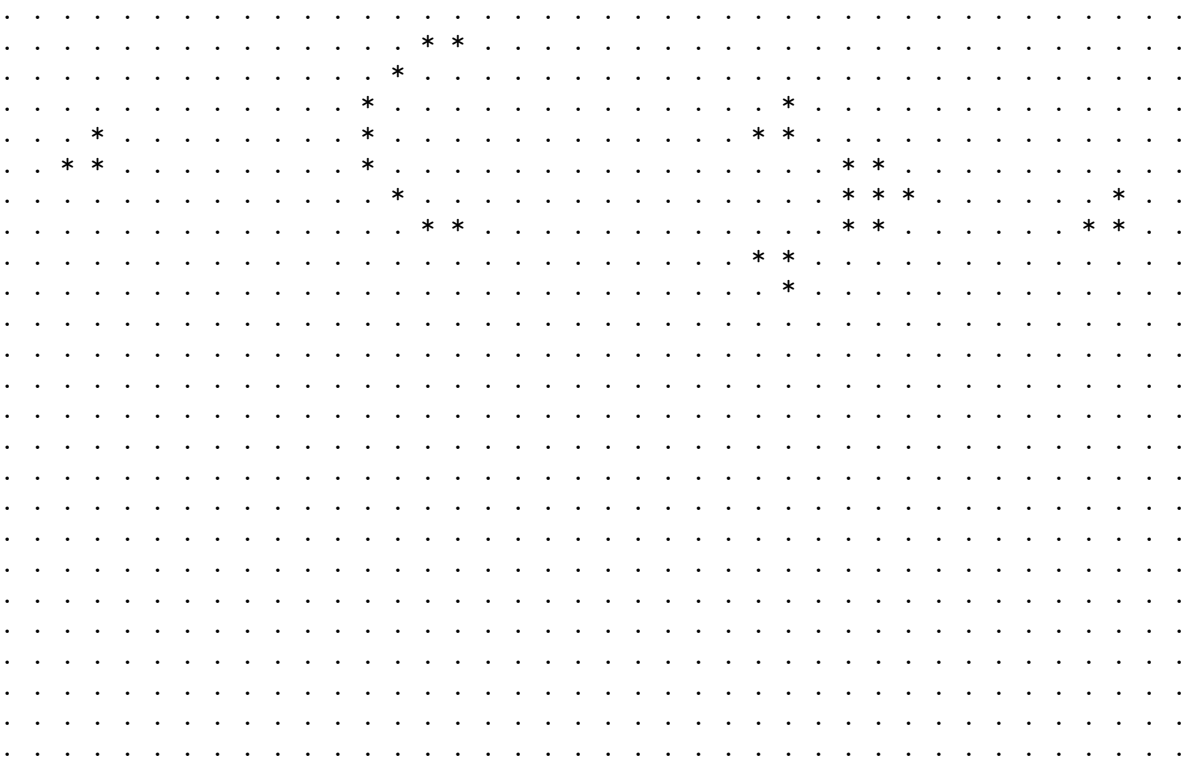
Example program flow:

```
Welcome to the Game of Life!
Load initial state from file? (y/n): n
Enter number of rows and columns, separated by a space: 5 5
. * * . *
* * * * *
* . . * .
. . . . *
. * * . .

Generation 0, live cells: 13
* . . . *
* . . . *
* . . . .
. * * * .
. . . . .

Generation 1, live cells: 8
. . . . .
* * . . .
* . * * .
. * * . .
. . * . .

Generation 2, live cells: 8
. . . . .
* * * . .
* . . * .
. . . . .
. * * . .
```

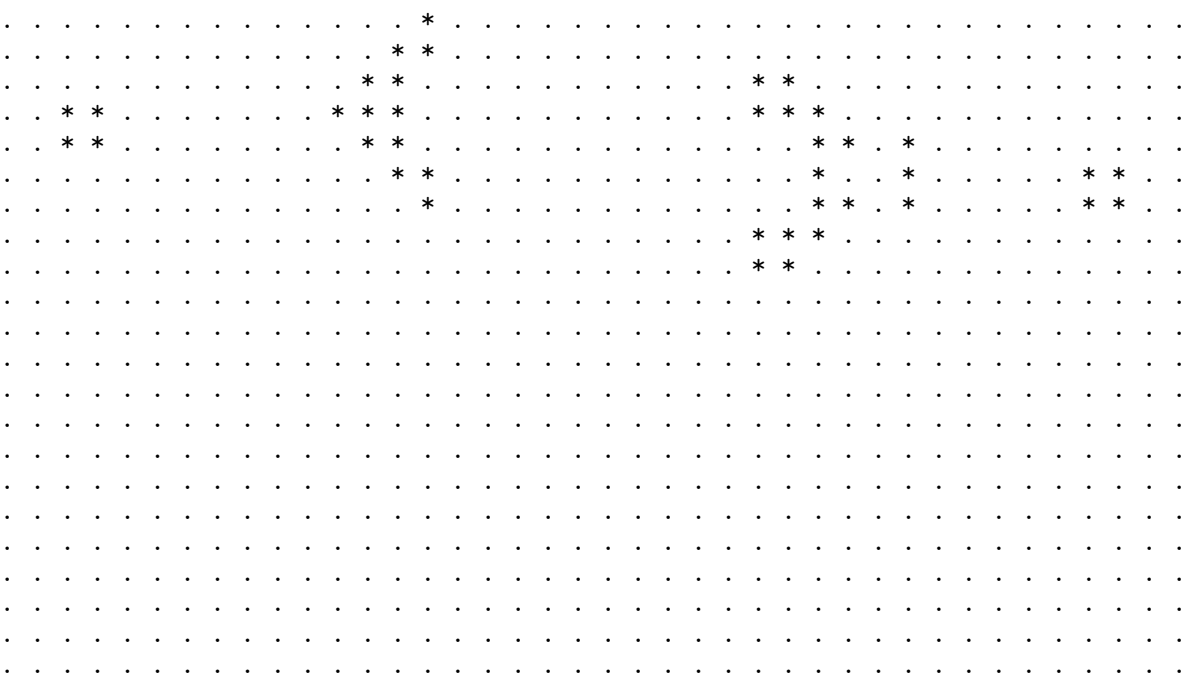
```
Generation 3, live cells: 7
. * . . .
* * * . .
* . * . .
. * * . .
. . . . .

Generation 4, live cells: 8
* * * . .
* . * . .
* . . * .
. * * . .
. . . . .

Generation 5, live cells: 9
* . * . .
* . * * .
* . . * .
. * * . .
. . . . .

Generation 6, live cells: 9
. . * * .
* . * * .
* . . * .
. * * . .
. . . . .

Generation 7, live cells: 9
. * * * .
. . . . *
* . . * .
. * * . .
. . . . .

Generation 8, live cells: 8
. . * * .
. * . . *
. * * * .
. * * . .
. . . . .

Generation 9, live cells: 9
Press ENTER to continue, 's' to save, or 'q' to quit: q
```

Example program flow:

```
Welcome to the Game of Life!
Load initial state from file? (y/n): y
Enter filename: initial_grid.txt
```

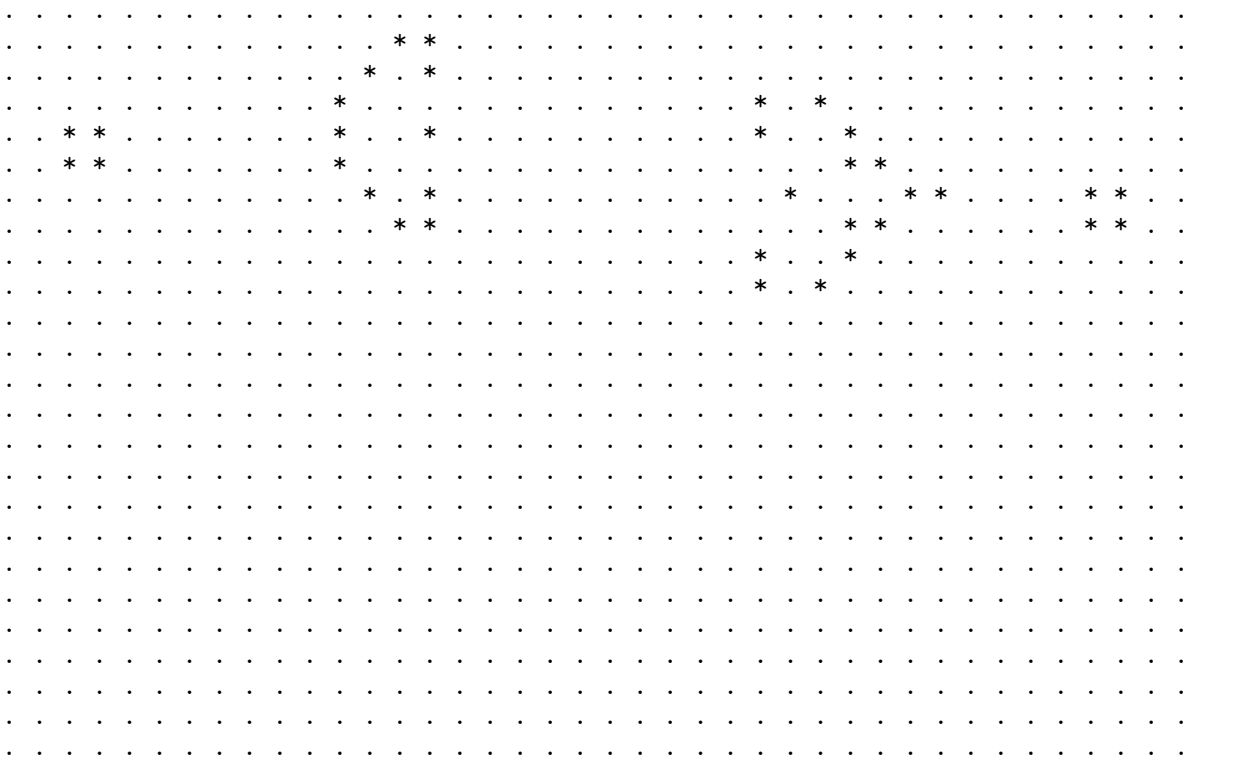


```
Generation 0, live cells: 28
```

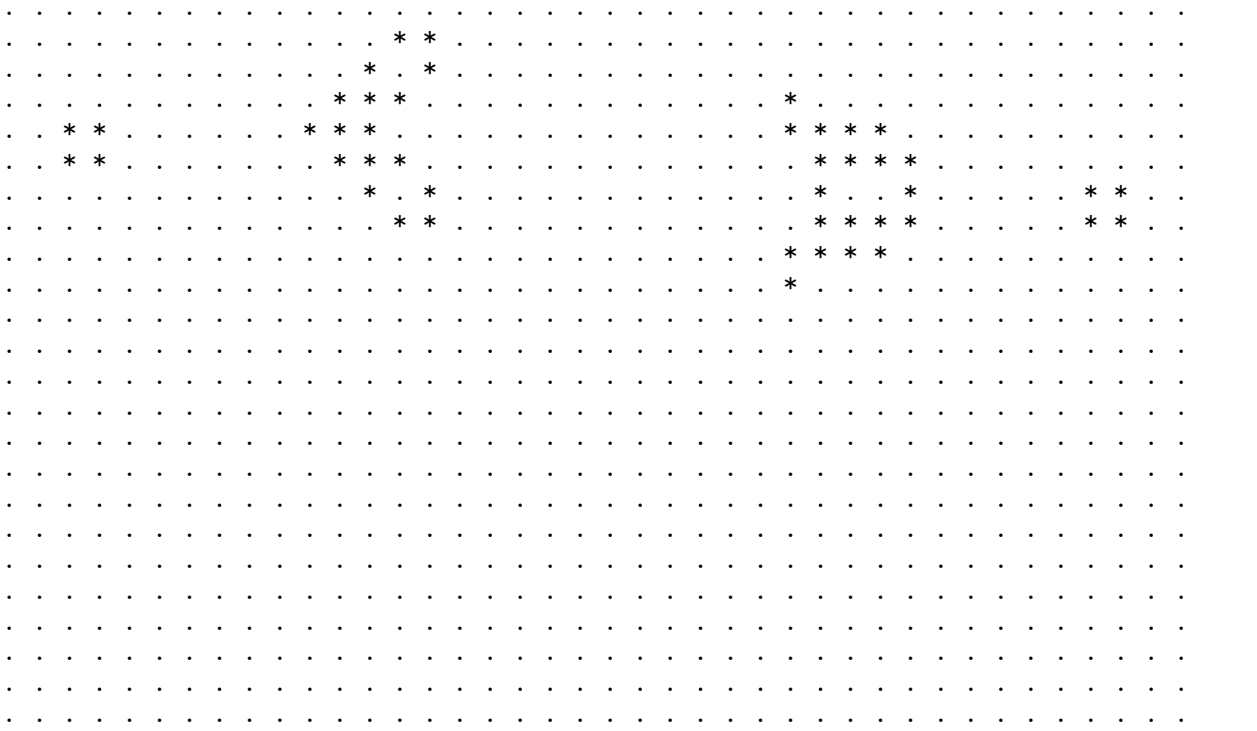


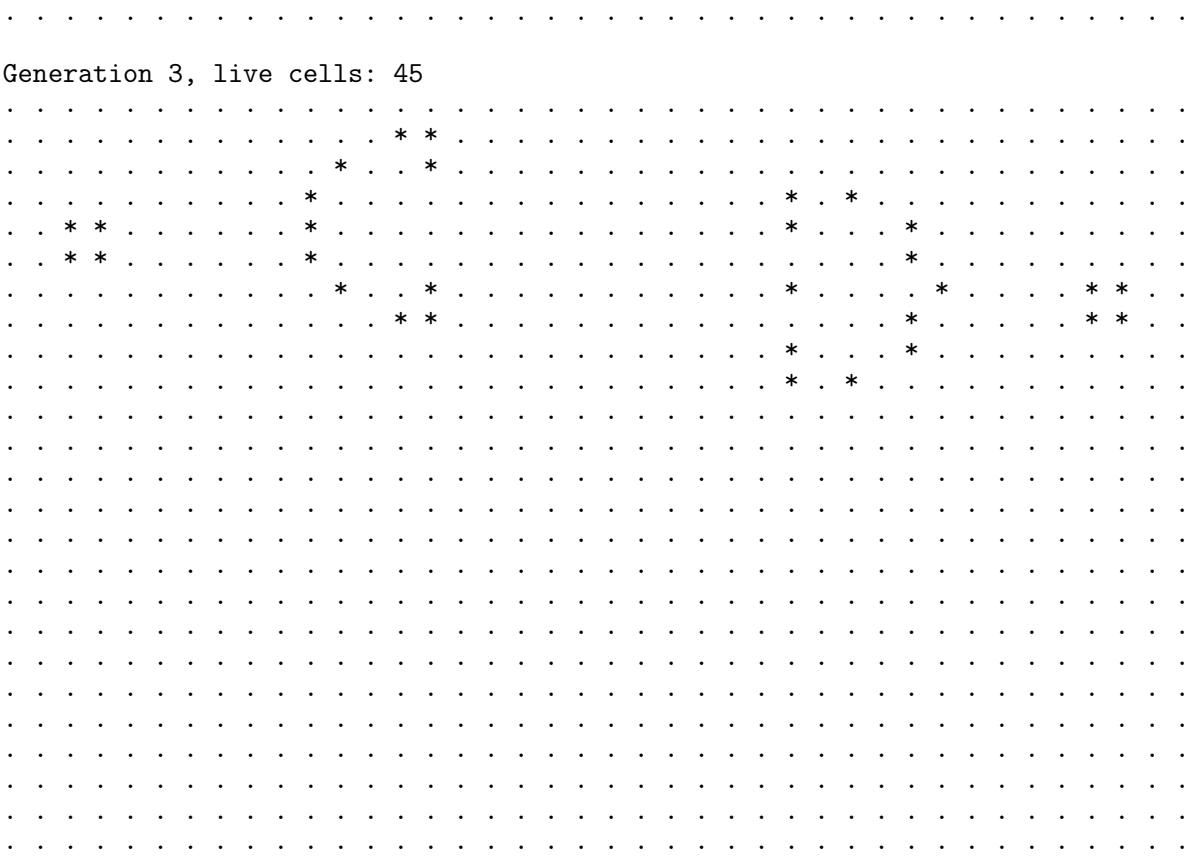
```
. . . . . .
. . . . . .
```

```
Generation 1, live cells: 39
```



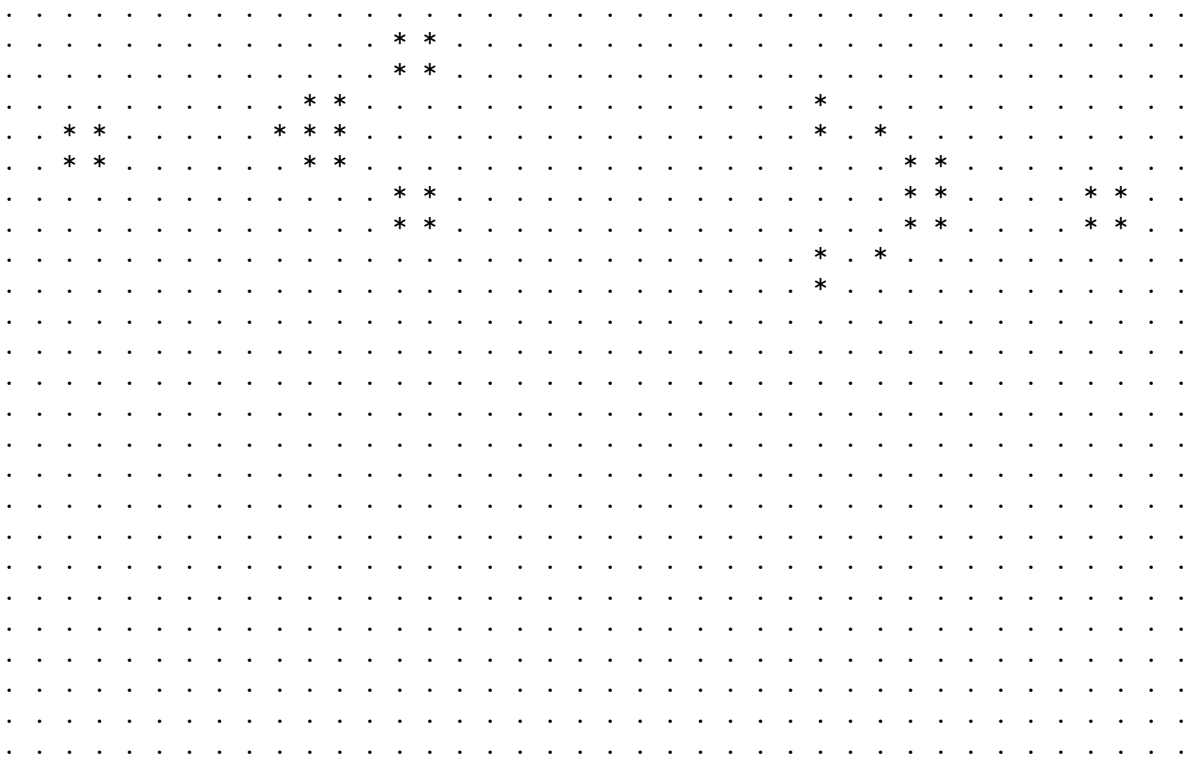
```
Generation 2, live cells: 35
```





Generation 3, live cells: 45

Generation 4, live cells: 31



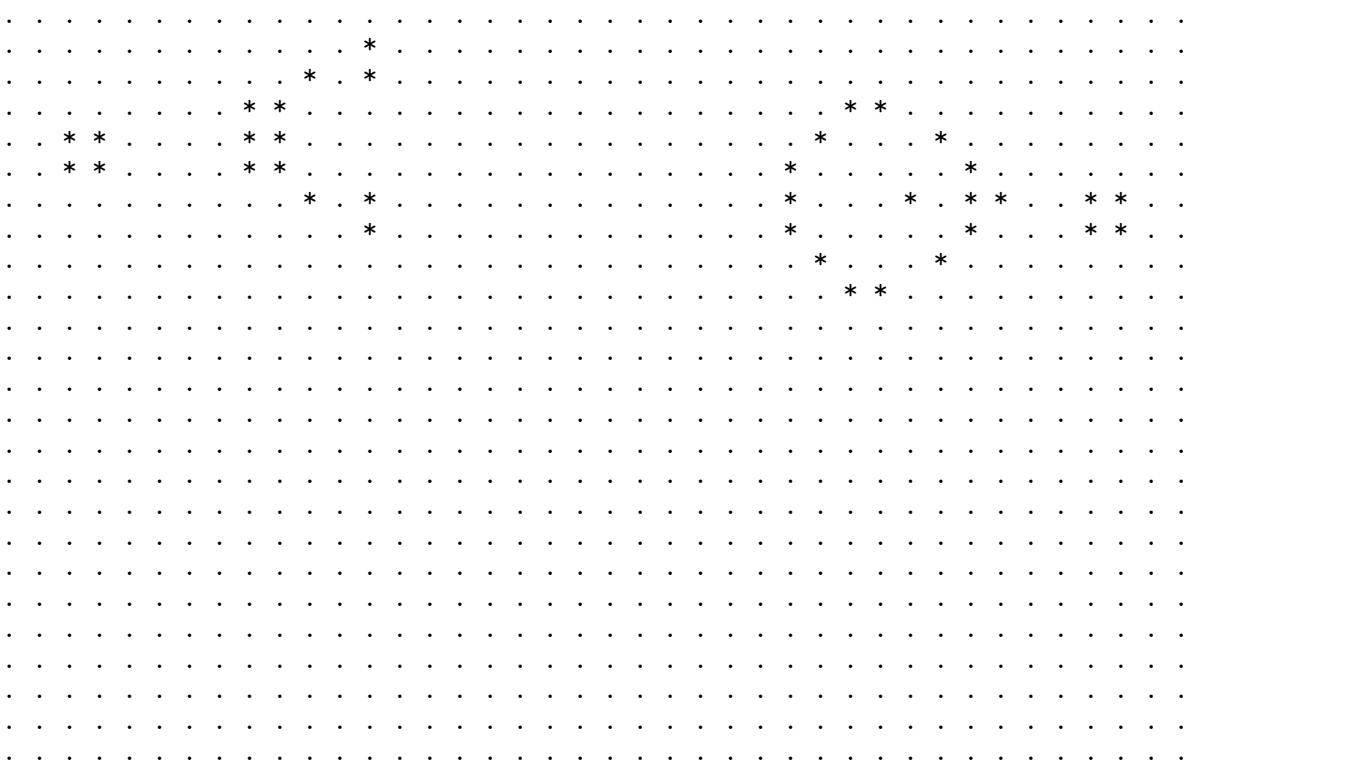
```
Generation 5, live cells: 35
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . * * . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . * * * . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . * . * * . . . . . . . . . . . . . . . * . . . . . . . . . . .
. . * * . . . . . * . . * . . . . . . . . . . . . . . . * . * . . . . . . . . .
. . * * . . . . . * . * * . . . . . . . . . . . . . . . . * . * . . . . . . . .
. . . . . . . . . . . . * * * . . . . . . . . . . . . . . * . . * . . . * * . .
. . . . . . . . . . . . . * * . . . . . . . . . . . . . . * . * . . . . * * . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . * . * . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . * . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Generation 6, live cells: 38
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . * . * . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . * . . * . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . * * . . . . . . . . . . . . . . . . . * . . . . . . . . . .
. . * * . . . . * * . . . * . . . . . . . . . . . . . . * . * . . . . . . . . .
. . * * . . . . . . * * . . . . . . . . . . . . . . . . * * . * . . . . . . . .
. . . . . . . . . . . * . . * . . . . . . . . . . . . . * * . * * . . . * * . .
. . . . . . . . . . . . * . * . . . . . . . . . . . . . * * . * . . . . * * . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . * . * . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . * . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

```
Generation 7, live cells: 39
. . . . . .
. . . . . * . . . . . .
. . . . . * * * * . . . . . .
. . . . . * * * * . . . . . * . . . . . .
. . * * . . . . * . . * . . . . . * . * . . . . . .
. . * * . . . . * * * * . . . . . * . . . * * . . . . . .
. . . . . * * * * . . . . . * . . . * * . . . * * . .
. . . . . * . . . * . . . . . * . . . * * . . . * * . .
. . . . . * . * . . . . . .
. . . . . * . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

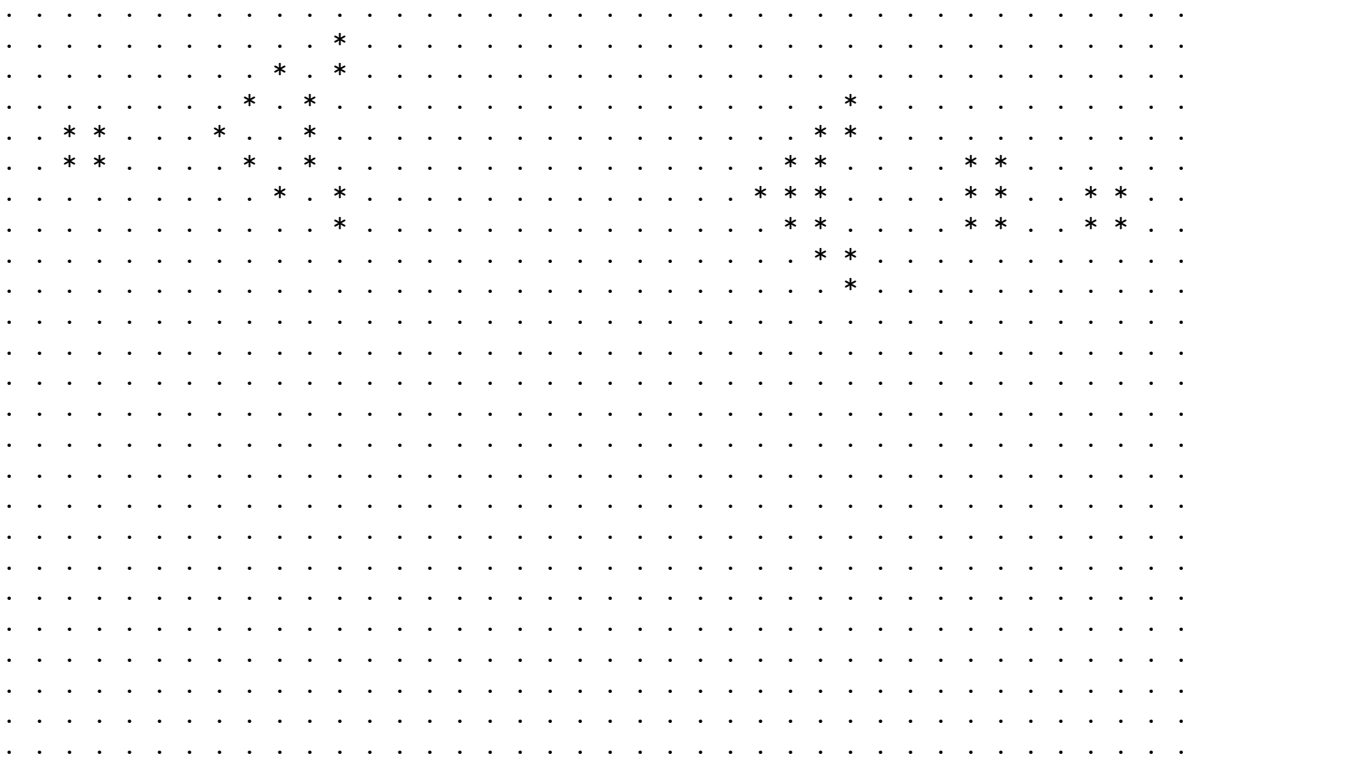
```
Generation 8, live cells: 43
. . . . . .
. . . . . * . * . . . . . .
. . . . . * . . . * . . . . . .
. . . . . * . . . . . * . . . . . .
. . * * . . . . * . . . . * . . . . . * * * . . . . . .
. . * * . . . . * . . . . . * * . * . * . . . . . .
. . . . . * . . . * . . . . . * * * . * . . * . . * * . .
. . . . . * . * . . . . . * * * . * . * . . . * * . .
. . . . . * * * * . . . . . .
. . . . . * . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

```
Generation 9, live cells: 43
```

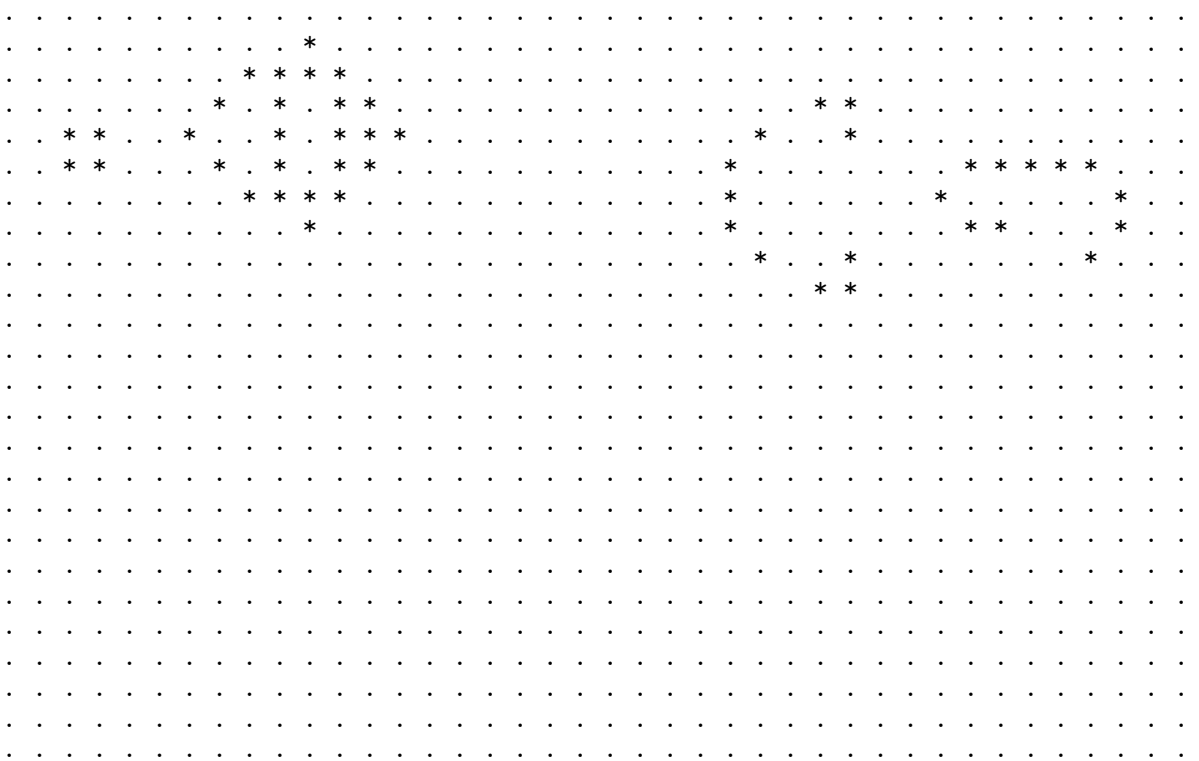
```
Press ENTER to continue, 's' to save, or 'q' to quit: s
Enter filename to save: new_grid.txt
```



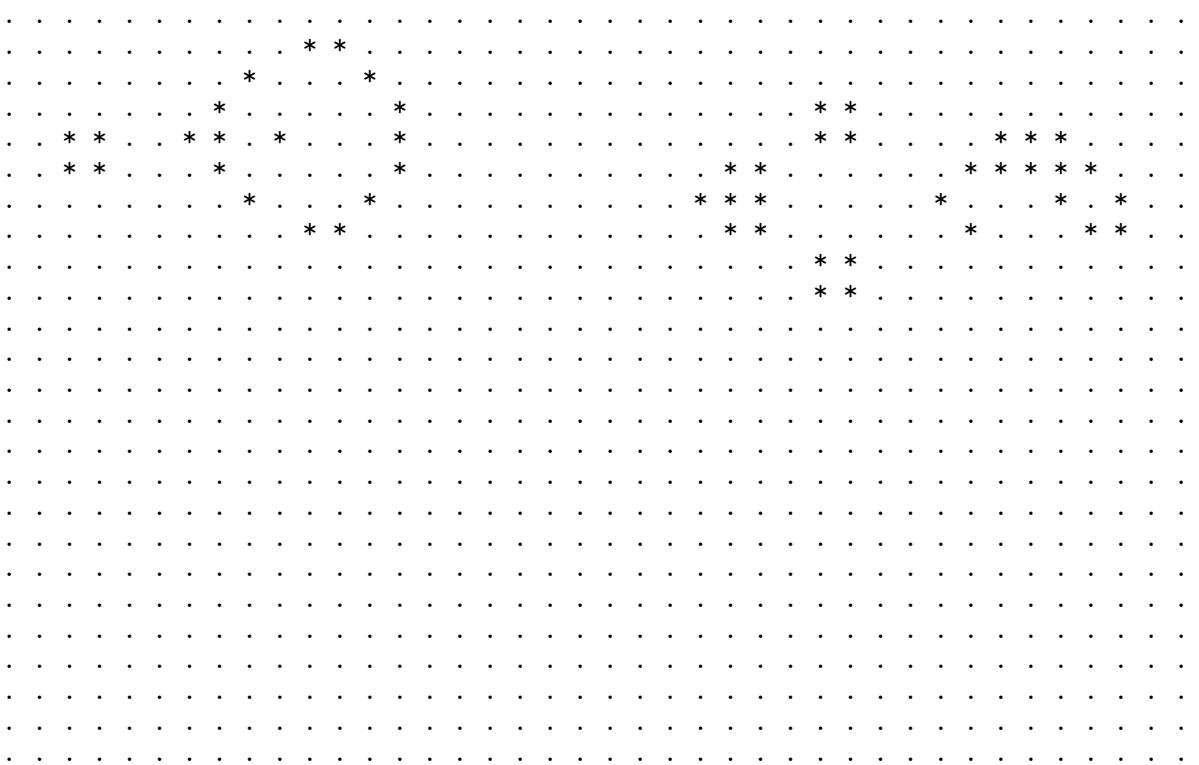
```
Generation 10, live cells: 36
```



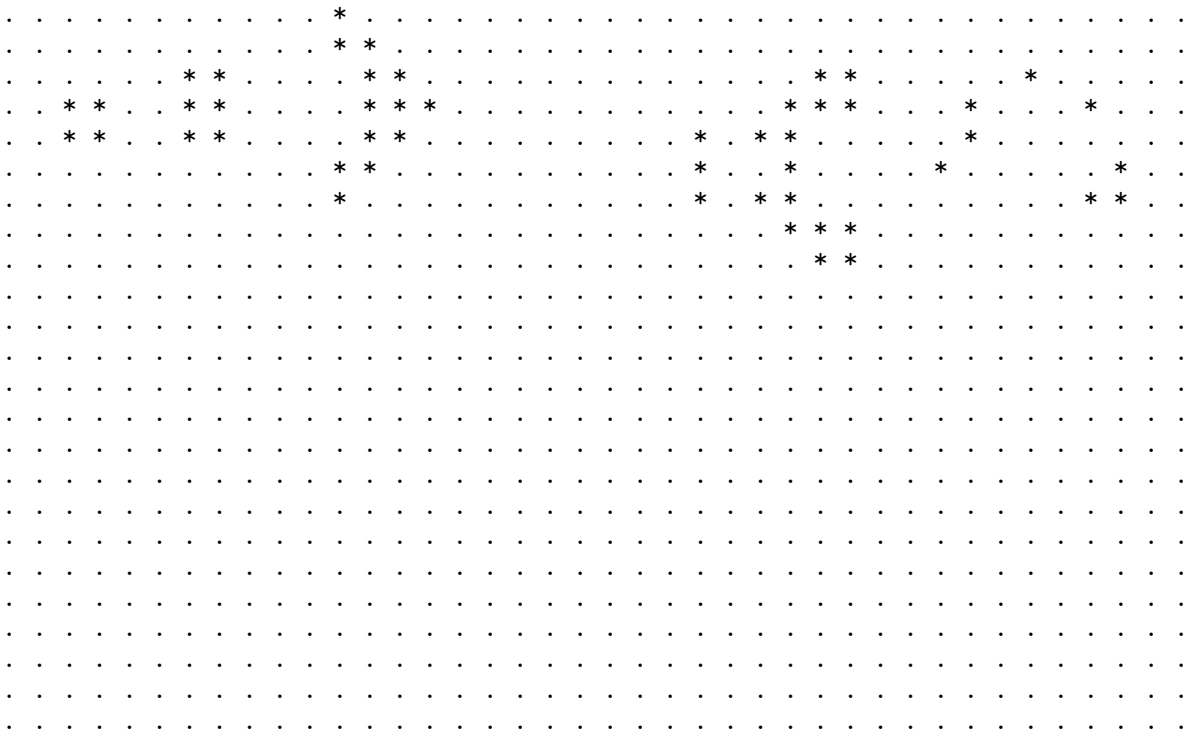
```
Generation 11, live cells: 39
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . * . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . * . * . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . * . * * . . . . . . . . . . . . . . . * * . . . . . . . . . . .
. . * * . . . * * . * * . . . . . . . . . . . . . . * . * . . . . . . . . . . .
. . * * . . . . * . * * . . . . . . . . . . . . . * . . . . . . * * . . . . . .
. . . . . . . . . * . * . . . . . . . . . . . . . * . . * . . * . . * . * * . .
. . . . . . . . . . * . . . . . . . . . . . . . . * . . . . . . * * . . * * . .
. . . . . . . . . . . . . . . . . . . . . . . . . . * . * . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . * * . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Generation 12, live cells: 42
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . * . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . * . * . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . * * . . . * . . . . . . . . . . . . . . * * . . . . . . . . . . .
. . * * . . . * * . . . * . . . . . . . . . . . . . * . * . . . . . . . . . . .
. . * * . . . * * . . . * . . . . . . . . . . . . * * * . . . . * * . . . . . .
. . . . . . . . . * . * . . . . . . . . . . . . * * * . . . . * . . * * * * . .
. . . . . . . . . . * . . . . . . . . . . . . . . * * * . . . . * * . * * * . .
. . . . . . . . . . . . . . . . . . . . . . . . . . * . * . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . * * . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Generation 13, live cells: 48
```



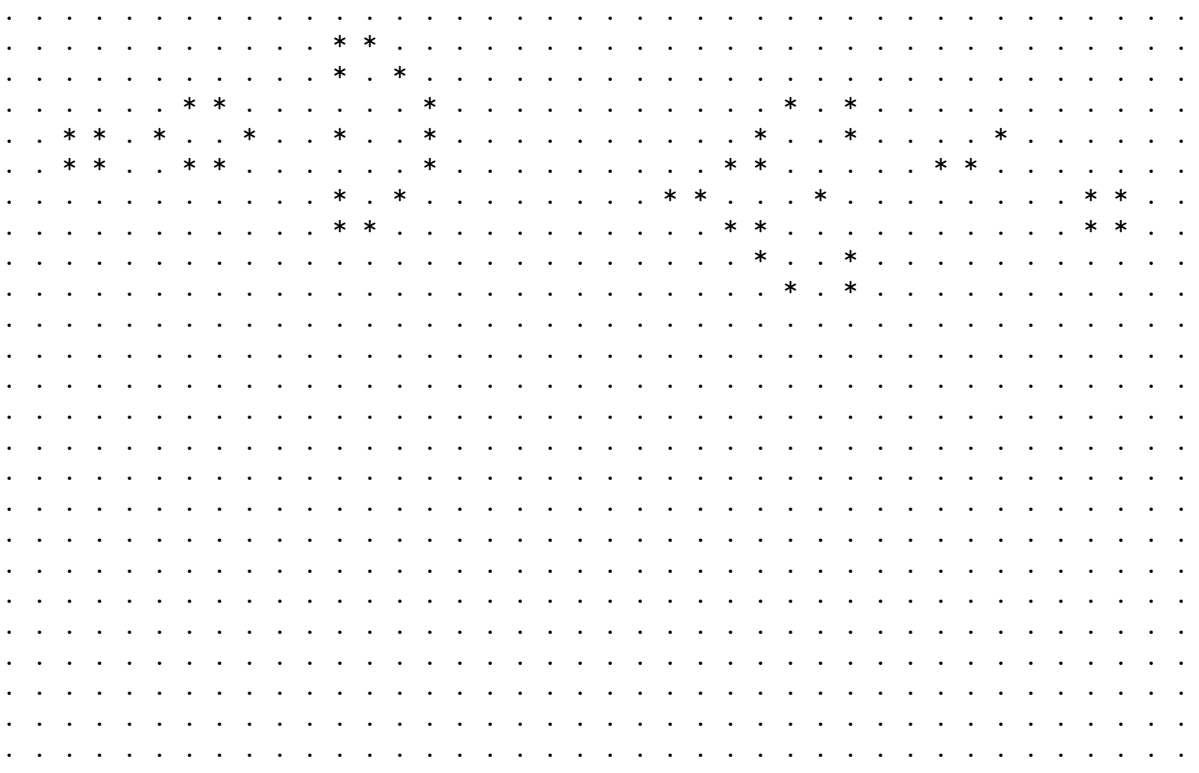
Generation 14, live cells: 49



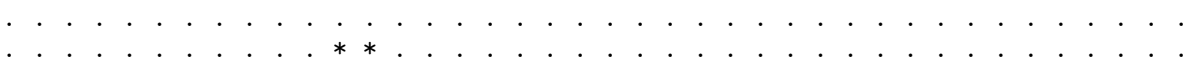
Generation 15, live cells: 49

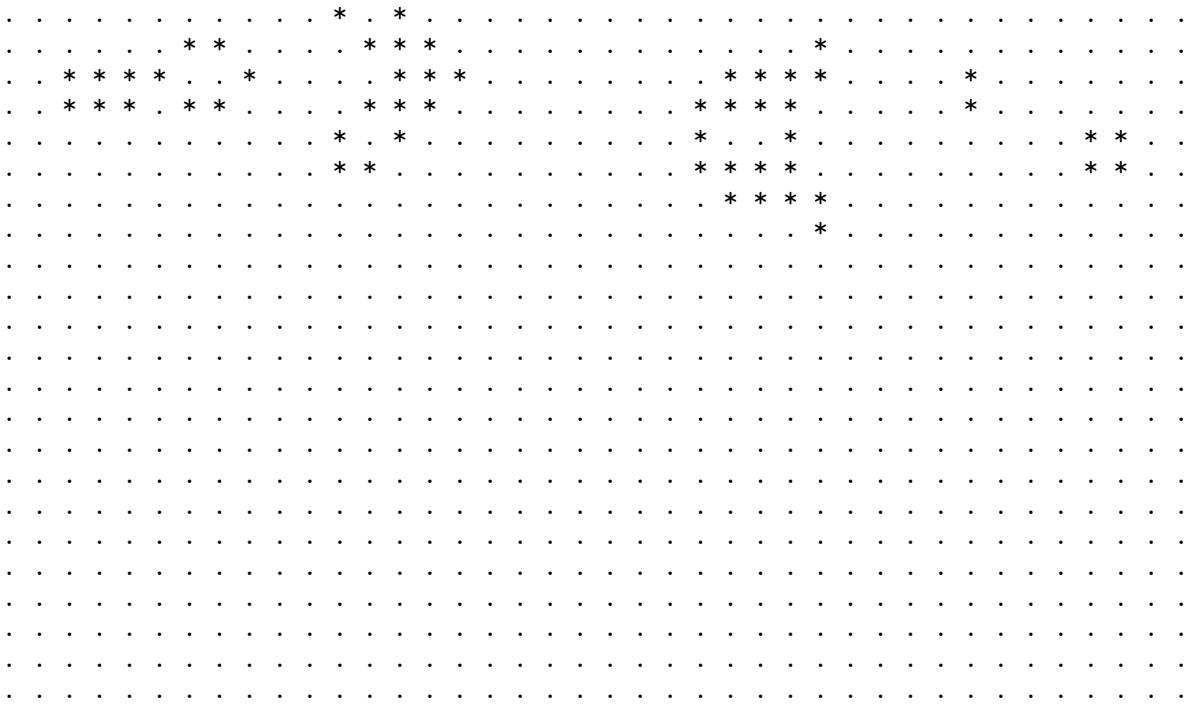


Generation 16, live cells: 49

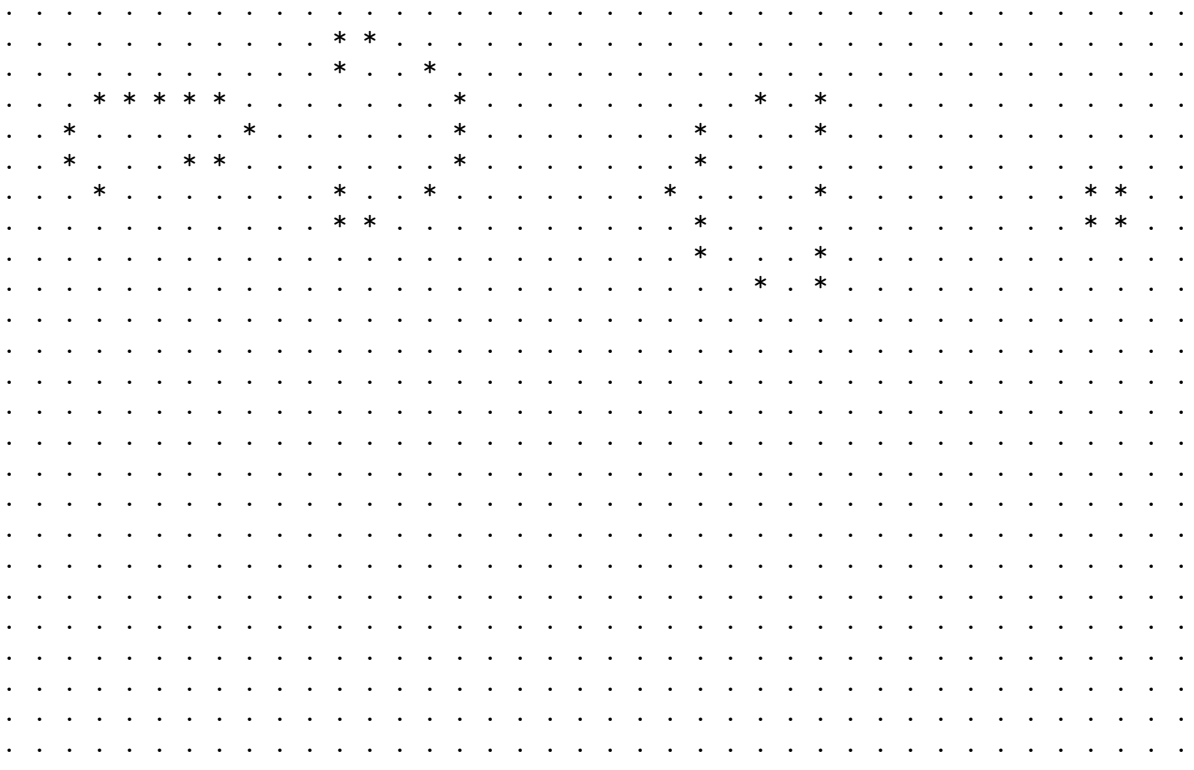


Generation 17, live cells: 44





```
Generation 18, live cells: 55
```



```
Generation 19, live cells: 38
Press ENTER to continue, 's' to save, or 'q' to quit: q
```