1. The PR department has decided to standardize how the brand name is used and will change Fluvip to FLUVIP in uppercase. Your task consists of run a Docker container using the image **fluvip/replace_text** hosted in DockerHub and change all the occurrences of "**fluvip**" and "**Fluvip**" to "**FLUVIP**" without affecting email addresses, these have to remain in lowercase i.e "someone@fluvip.com" must **NOT** be changed to "someone@FLUVIP.com". When you complete the task, please attach the lines used to solve the problem.

. The Game of Life**:** "*The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead, or "populated" or "unpopulated". Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:*

- *Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.*
- *Any live cell with two or three live neighbours lives on to the next generation.*
- *Any live cell with more than three live neighbours dies, as if by overpopulation.*
- *Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.*

*The initial pattern constitutes the seed of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a tick (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.*" As it appears in https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Following a BDD paradigm, write the scenarios for 4 rules of the game. This exercice does not require any implementation, only the scenarios.

3. Continuing with the previous exercise, write the production code to make the scenarios pass, taking into account the following restrictions:

- At the end of the exercice all test must pass
- The final code must NOT contain any "if", "else", "switch", "case", "unless". The idea here is to implement the rules without conditionals
- The final code must NOT contain any "while", "loop", "do", if you implement the solution in Ruby, the word "do" is only allowed for blocks not for loops.

The implementation is required only for the rules of the game, no rendering is required.

4. Suppose you have the following database.

| Datos | | |
|---|---|---|
| Cliente | Variable | Valor |
| 1 | Genero | M |
| 2 | Genero | M |
| 3 | Genero | F |
| 4 | Genero | M |
| 5 | Genero | M |
| 6 | Genero | M |
| 7 | Genero | F |
| 8 | Genero | M |
| 9 | Genero | F |
| 1 | Ciudad | Bogota |
| 2 | Ciudad | Cali |
| 3 | Ciudad | Bogota |
| 4 | Ciudad | Medellin |
| 5 | Ciudad | Medellin |
| 6 | Ciudad | Barranquilla |
| 7 | Ciudad | Cali |
| 8 | Ciudad | Bogota |
| 9 | Ciudad | Medellin |
| 1 | Mascota | Si |
| 2 | Mascota | Si |
| 3 | Mascota | Si |
| 4 | Mascota | No |
| 5 | Mascota | Si |
| 6 | Mascota | No |
| 7 | Mascota | No |
| 8 | Mascota | No |
| 9 | Mascota | Si |

| Clientes | |
|---|---|
| ClienteID | Nombre |
| 1 | Juan |
| 2 | Pepe |
| 3 | Maria |
| 4 | Jose |
| 5 | David |
| 6 | Jaime |
| 7 | Diana |
| 8 | Ivan |
| 9 | Cata |

Write a SQL query to retrieve the names of women who have pets in Bogota. Write the names that the query would return.