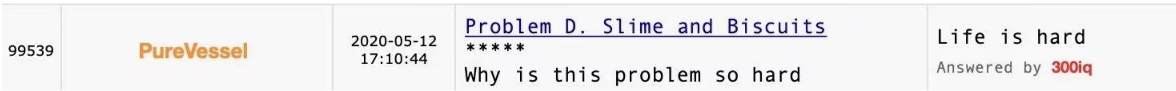
Life is Hard and Undecidable, but...

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 1024 megabytes

To John Horton Conway, his Game of Life, and also the life we are living...

Disclaimer: For those who have no patience for the whole context or are already familiar with Game of Life, the problem is in the last paragraph on the third page.

Life is hard. There are beautiful dreams, yet you can never realize them. There are regrets and lost loved ones, yet you can never turn back time. There is death lying ahead with the anxiety it persistently gives you, yet you have no way to escape from it. There are problems right here for you to solve, yet you simply don't know how.



Life is hard.

Life is undecidable. You just don't know what you are going to get from this box of chocolates. Experienced through all the Black Swan and Grey Rhino events happening those years, we all wonder, what's going to happen next? What will the future be like? *"C'est la vie", say the old folks. It goes to show you never can tell.*

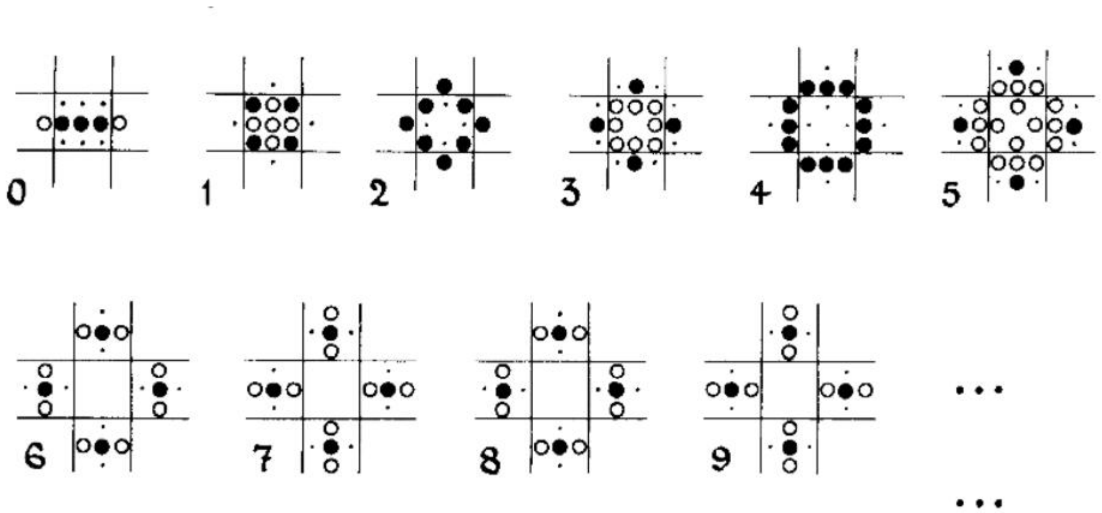
But, life just goes on. We have wines to drink, songs to sing, problems to solve, and games to play. The game we will play now is, **Game of Life**, or simply, **Life**, devised by the famous British mathematician John Horton Conway in 1970.

Life is a zero-player "game" played on an **infinite** squared board. At any time, some of the cells will be **live** and others **dead**. Which cells are live at generation 0 is up to you! But then you've got nothing else to do because the state at any later time is determined by the previous one by the following rules:

- **BIRTH.** A cell that's *dead* at generation t becomes *live* at generation $t+1$ if and only if *exactly three* of its eight neighbors were live at generation t.
- **DEATH by overcrowding.** A cell that's *live* at generation t becomes *dead* at generation $t+1$ if *at least four* of its eight neighbors were live at generation t.
- **DEATH by exposure.** A cell that's *live* at generation t becomes *dead* at generation $t+1$ if *at most one* of its eight neighbors were live at generation t.
- **Survival.** A cell that's *live* at generation t is still *live* at generation $t+1$ if *two or three* of its eight neighbors were live at generation t.

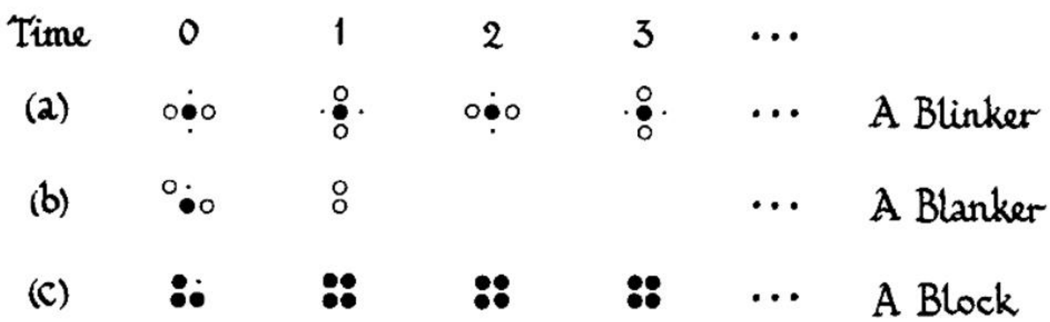
The rule above can be summarized and memorized as "Just 3 for birth, 2 or 3 for survival."

The following picture is an example of starting from a configuration where only five cells in a line are live. This will end up forming a "traffic light" pattern that repeats **infinitely** with a period of 2. In the picture, filled circles represent *live* cells that will still be *live* at the next generation. Unfilled circles represent *live* cells that will be *dead* at the next generation. Dots represent *dead* cells that will be *live* at the next generation. (Credit: All pictures below are from *Winning Ways For Your Mathematical Plays, Volume 4* by Elwyn R. Berlekamp and John H. Conway).



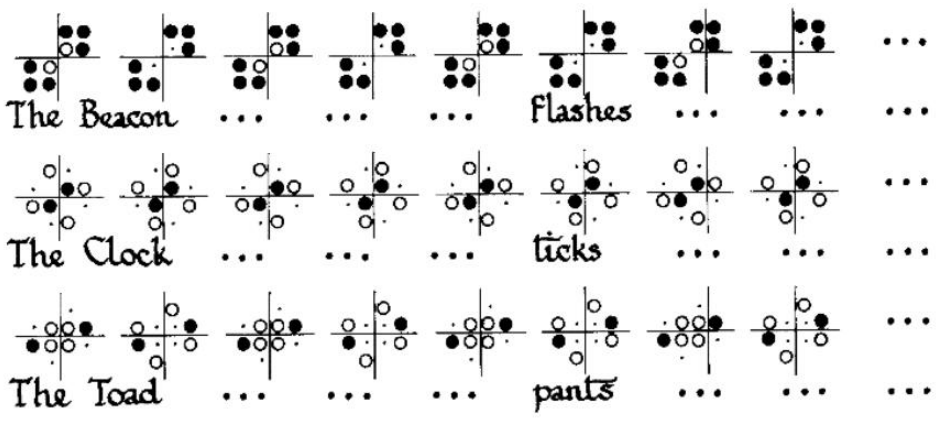
A Line of five becomes traffic lights.

There are some other examples of starting configurations with different “destinies":



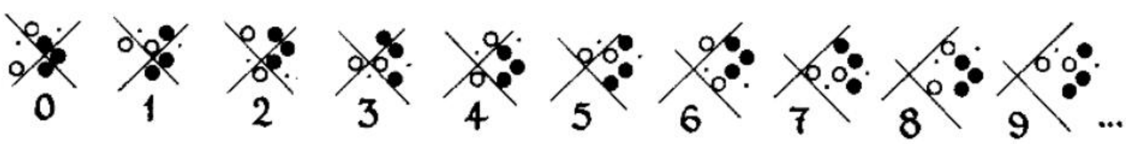
Different start, different destiny.

There are examples where one ends up in a life cycle,



Three life cycles with period 2.

and an example of a moving “glider".



A glider moving diagonally.

Mathematicians and theoretical computer scientists are fascinated by the mystery of Life. They ask the following question: "Given a starting configuration of Life, can we decide if it will eventually die out? i.e., is there some finite number t such that there is no live cells at generation t?" However, it turns out this problem is not only **NP-hard**, but even **undecidable**. It is possible to build a pattern in Life that acts like a finite-state machine connected to two counters. This has the same computational power as a universal Turing machine, so Life is theoretically as powerful as any computer with unlimited memory and no time constraints; it is Turing complete. In fact, several different programmable computer architectures have been implemented in Life, including a pattern that simulates Tetris.

Although as the above suggests, we can never unravel the mystery of Life, there are still certain things we can do. Construction is just one of those things, and is also exactly what this problem asks for:

> Given a positive integer k ($1 \le k \le 100$), construct a starting configuration of Game of Life, with all live cells having **positive** coordinates not exceeding 300, so that the configuration lasts **exactly** k generations, that is, there exist live cells at generation $k-1$, but no live cells at generation k. It is guaranteed that at least one such starting configuration exists.

Input

The input consists of only one integer k ($1 \le k \le 100$), denoting the number of generations the configuration need to last.

Output

The first line contains an integer n, ($1 \le n \le 90000$), denoting the number of live cells in your starting configuration.

Then n lines follow, each line containing two integers x, y ($1 \le x, y \le 300$), denoting the coordinates of a live cell in your starting configuration. The coordinates of each live cell should be output exactly once.

If there are many possible starting configurations, outputting any of them will be considered correct.

Examples

standard input	standard output
1	1 1 1
2	3 100 100 101 100 102 99