# Problem K. Circle of Life

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Your friend Lucky is a little elf. Recently, Lucky discovered a magical creature, she named them "Twinkle". Lucky has a magical chain of $n$ vertices and $n-1$ edges, where the vertices are connected by the edges one by one. We assume that the vertices are numbered 1 to $n$ from left to right, and the $i$-th edge connects vertex $i$ and vertex $i+1$. She found that Twinkles could live in the magical chain.

Lucky can put some Twinkles on some vertices simultaneously with her magic, but there should be at most one Twinkle in one vertex since the near Twinkles will perish together. Then, every second, each Twinkle will split into two Twinkles simultaneously and they will dash in the opposite directions to the two adjacent vertices.

Formally, a Twinkle in vertex $u$ splits into two Twinkles, the left splited Twinkle and the right splited Twinkle.

- The left splited Twinkle will dash toward the left and reach the vertex $u-1$.
- The right splited Twinkle will dash toward the right and reach the vertex $u+1$.

But unluckily, if one side has no vertex, the Twinkle will dash out of the chain and die out(for the left splited Twinkle in vertex 1 and the right splited Twinkle in vertex $n$). Much more unfortunately, if two Twinkles dash to have a head-on collision (i.e. meet in the same vertex or edge), they will perish together with a tearing crash! Specifically, a crash will happen in two situations:

1. The right splited Twinkle in vertex $i$ and the left splited Twinkle in vertex $i+2$ will crash in vertex $i+1$(assuming that vertex $i+1$ lives no Twinkle).
2. The right splited Twinkle in vertex $i$ and the left splited Twinkle in vertex $i+1$ will crash in the $i$-th edge.

Lucky hopes there's always some Twinkle living on the chain. In addition, in order to be more convenient for Lucky to check the validity, there should be duplicated configurations within $2n$ seconds. Specifically, a configuration can be denoted by a binary string $S$ of length $n$, where $S_i = 1$ iff vertex $i$ lives a Twinkle, and $C_i$ denotes the configuration after $i$ seconds. Your task is to find an initial configuration $C_0$ so that for each $i\,(0 \le i \le 2n), C_i \neq 00\cdots00$ and that there exist two integers $i, j\,(0 \le i < j \le 2n), C_i = C_j$.

## Input

Input one line containing one integer $n\,(2 \le n \le 123)$, denoting the length of the chain.

## Output

If there is no solution, just output "Unlucky"(without quotes) in one line.

Otherwise, output one line containing one binary string, denoting the initial configuration you found.

## Examples

| standard input | standard output |
|---|---|
| 2 | 10 |
| 4 | 1000 |

## Note

For the first case, the configurations will change like this:

$$10 \to 01 \to 10$$

, where $C_0 = C_2$ holds.

For the second case, the configurations will change like this:

$$1000 \to 0100 \to 1010 \to 0001 \to 0010 \to 0101 \to 1000$$

, where $C_0 = C_6$ holds.

# Problem L. Three,Three,Three

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |



A poor little penguin has been thrown by a killer whale for a distance of one hundred and eight thousand li. The poor little penguin found itself on a mysterious island then. On the ground there are many graphs. The poor little penguin found that all the vertices are of degree three. "The Dao produced One; One produced Two; Two produced Three; Three produced All things", the poor little penguin recalled this mystical sentence. Suddenly, a quill-pen came into view. A sound said, "Every time you dip the quill-pen in the sea, you can one-touch-draw three edges. That is, you can start at a vertex, pass three different edges to a vertex." It is a parrot with three golden feathers on top of its head, "If all edges of a graph could be drawn once and only once by this way, then we call it a Three-throwable graph. You should find all the Three-throwable graphs, and show me how to throw three! As soon as you're done, I will bestow you the Produced-All-Things Power and send you back to avenge yourself on the killer whale!"

Too many graphs! So the poor little penguin asks you three for help. Please help it, for Light and Justice.

In short, given a graph of $n$ vertices and $m$ edges, where $n$ is even and all the vertices are of degree 3. So it can be seen that $m = \frac{3n}{2}$ and that $m$ is multiple of 3. You should try to split the graph into $\frac{m}{3}$ chains of length 3. Determine if the graph is Three-throwable graph(such chain spliting scheme exists) and output the splited chains if solution exists.

## Input

The first line contains two integers $n, m\,(2 \le n \le 500)$, denoting the number of vertices and edges.

Following $m$ lines each contains two integers $u, v\,(1 \le u, v \le n)$, denoting an edge connecting vertex $u$ and $v$.

It is guaranteed that $n$ is even, $m = \frac{3n}{2}$ and that all the vertices are of degree 3.

## Output

If it is not a Three-throwable graph, output "IMPOSSIBLE"(without quotes) in one line.

Otherwise, output $\frac{m}{3}$ lines each contains four integers $a_i, b_i, c_i, d_i$, denoting one chain containing three edges $a_i \leftrightarrow b_i, b_i \leftrightarrow c_i, c_i \leftrightarrow d_i$. The multiset of given edges and the multiset of all the chain edges should be the same. If multiple solutions, output any one of them.

## Examples

| standard input | standard output |
| --- | --- |
| 2 3<br>1 2<br>1 2<br>1 2 | 2 1 2 1 |
| 4 6<br>1 1<br>1 2<br>2 3<br>2 3<br>3 4<br>4 4 | 1 1 2 3<br>2 3 4 4 |
| 4 6<br>1 1<br>2 2<br>3 3<br>1 4<br>2 4<br>3 4 | IMPOSSIBLE |