

# 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 splitted Twinkle and the right splitted Twinkle.

- The left splitted Twinkle will dash toward the left and reach the vertex  $u - 1$ .
- The right splitted 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 splitted Twinkle in vertex 1 and the right splitted 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 splitted Twinkle in vertex  $i$  and the left splitted Twinkle in vertex  $i + 2$  will crash in vertex  $i + 1$ (assuming that vertex  $i + 1$  lives no Twinkle).
2. The right splitted Twinkle in vertex  $i$  and the left splitted 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  $i$  ( $0 \leq i \leq 2n$ ),  $C_i \neq 00 \cdots 00$  and that there exist two integers  $i, j$  ( $0 \leq i < j \leq 2n$ ),  $C_i = C_j$ .

## Input

Input one line containing one integer  $n$  ( $2 \leq n \leq 123$ ), 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:

$$10 \rightarrow 01 \rightarrow 10$$

, where  $C_0 = C_2$  holds.

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

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

, where  $C_0 = C_6$  holds.