Many LCS

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Given K, construct two non-empty **binary** strings S and T of length **at most 8848** such that they have exactly K different longest common subsequences. More formally, if L is the length of the longest common subsequence of S and T, there should exist exactly K distinct binary strings of length L which are subsequences of both S and T.

It is guaranteed that under the constraints of this problem such strings always exist.

Input

The only line of input contains a single integer K ($1 \le K \le 10^9$).

Output

Print non-empty binary strings S and T on separate lines. The length of each of them should not exceed 8848. They can have different lengths.

If there is more than one solution, you can print any one of them.

Examples

standard input	standard output
1	1111 00
2	10 01
3	010 1001
100	10001000001011100001010100011 11000010001010101001010011100

Note

In the first example, the longest common subsequence of `1111` and `00` has length 0, and there exists only one string of length 0 which is a subsequence of both of them — the empty string.

In the second example, the length of the longest common subsequence of strings `10` and `01` is 1, and there are 2 strings of length 1 which are subsequences of both S and T: `0` and `1`.

In the second example, the length of the longest common subsequence of strings `010` and `1001` is 2, and there are 3 strings of length 2 which are subsequences of both S and T: `00`, `01`, and `10`.

It would be disrespectful to make strings longer than Everest...