A bitstring, whose length is one less than a prime, might be **magic**. `1001` is one such string. In order to see the **magic** in the string let us append a non-bit `x` to it, regard the new *thingy* as a cyclic string, and make this square matrix of bits

each bit	`1001`
every 2^{nd} bit	`0110`
every 3^{rd} bit	`0110`
every 4^{th} bit	`1001`

This matrix has the same number of rows as the length of the original bitstring. The m-th row of the matrix has every m-th bit of the original string starting with the m-th bit. Because the enlarged *thingy* has prime length, the appended `x` never gets used.

If each row of the matrix is either the original bitstring or its complement, the original bitstring is **magic**.

Input

Each line of input (except last) contains a prime number $p \leq 100000$. The last line contains '`0`' and this line should not be processed.

Output

For each prime number from the input produce one line of output containing the lexicographically smallest, non-constant **magic** bitstring of length $p - 1$, if such a string exists, otherwise output '`Impossible`'.

Sample Input

```
5
3
17
47
2
79
0
```

Sample Output

```
0110
01
0010111001110100
000010000110101000110110010011101010011110111 1
Impossible
00100110000101101000000100111100111010101010001100001101111110100101111001101 1
```