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 $2^{\text {nd }}$ bit | 0110 |
| every $3^{\text {rd }}$ bit | 0110 |
| every $4^{t h}$ 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

01010100101110000011110010010000101 00111001000110101110111101101000100 00111010110000100000100101010110001 00100111111010010010000101000100011 11100111111010011111001001010111010 00100111011001000101101010100100100 00000010000110001000100111100100001 10001100000111010111111001100001001 00110101000010011100010111010001101 11010110100111110111111011000011110 10101000001100011000100111111101100 00101110000111111011011010010000001 11001111011001110110000111010110010 10101010101011001010001111001000110 01000011000111111011010010010000001 11100010111100100000001101110011100 11111010101000011110010000001000001 10100101000100111010001011100011011 11010100110110111100110000001010001 11110011100111101100001101110111001 11101111111101101101010100101110110 01000110111010001010110110000011010 00000110000011101110101111011011010 00000110110111001010101101111101111 00101000111101110100100001000101001 11011000110101111011011000011111000 10110101010011101001100011110100100 1110011011 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
0000100001101010001101100100111010100111101111
Impossible
001001100001011010000001001111001110101010100011000011011111101001011110011011

