

## B. Blackboard Fibonacci

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Fibonacci numbers are the sequence of integers:  $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2, f_4 = 3, f_5 = 5, \dots, f_n = f_{n-2} + f_{n-1}$ . So every next number is the sum of the previous two.

Bajtek has developed a nice way to compute Fibonacci numbers on a blackboard. First, he writes a 0. Then, below it, he writes a 1. Then he performs the following two operations:

- operation "T": replace the top number with the sum of both numbers;
- operation "B": replace the bottom number with the sum of both numbers.

If he performs  $n$  operations, starting with "T" and then choosing operations alternately (so that the sequence of operations looks like "TBTBTBTB..."), the last number written will be equal to  $f_{n+1}$ .

Unfortunately, Bajtek sometimes makes mistakes and repeats an operation two or more times in a row. For example, if Bajtek wanted to compute  $f_7$ , then he would want to do  $n = 6$  operations: "TBTBTB". If he instead performs the sequence of operations "TTTBBT", then he will have made 3 mistakes, and he will incorrectly compute that the seventh Fibonacci number is 10. The number of mistakes in the sequence of operations is the number of neighbouring equal operations («TT» or «BB»).

You are given the number  $n$  of operations that Bajtek has made in an attempt to compute  $f_{n+1}$  and the number  $r$  that is the result of his computations (that is last written number). Find the minimum possible number of mistakes that Bajtek must have made and any possible sequence of  $n$  operations resulting in  $r$  with that number of mistakes.

Assume that Bajtek always correctly starts with operation "T".

### Input

The first line contains the integers  $n$  and  $r$  ( $1 \leq n, r \leq 10^6$ ).

### Output

The first line of the output should contain one number — the minimum possible number of mistakes made by Bajtek.

The second line should contain  $n$  characters, starting with "T", describing one possible sequence of operations with that number of mistakes. Each character must be either "T" or "B".

If the required sequence doesn't exist, output "IMPOSSIBLE" (without quotes).

### Examples

input
6 10
output
2 TBBTTB

input
4 5
output
0 TBTB

input

2 1

output

IMPOSSIBLE