

B. Friends

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

One day Igor K. stopped programming and took up math. One late autumn evening he was sitting at a table reading a book and thinking about something.

The following statement caught his attention: "Among any six people there are either three pairwise acquainted people or three pairwise unacquainted people"

Igor just couldn't get why the required minimum is 6 people. "Well, that's the same for five people, too!" — he kept on repeating in his mind. — "Let's take, say, Max, Ilya, Vova — here, they all know each other! And now let's add Dima and Oleg to Vova — none of them is acquainted with each other! Now, that math is just rubbish!"

Igor K. took 5 friends of his and wrote down who of them is friends with whom. Now he wants to check whether it is true for the five people that among them there are either three pairwise acquainted or three pairwise not acquainted people.

Input

The first line contains an integer m ($0 \leq m \leq 10$), which is the number of relations of acquaintances among the five friends of Igor's.

Each of the following m lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq 5; a_i \neq b_i$), where (a_i, b_i) is a pair of acquainted people. It is guaranteed that each pair of the acquaintances is described exactly once. The acquaintance relation is symmetrical, i.e. if x is acquainted with y , then y is also acquainted with x .

Output

Print "FAIL", if among those five people there are no either three pairwise acquainted or three pairwise unacquainted people. Otherwise print "WIN".

Examples

input
4 1 3 2 3 1 4 5 3
output
WIN

input
5 1 2 2 3 3 4 4 5 5 1
output
FAIL