The Emperor

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Note: This is the inverse version of problem "The Empress" with differences in constraints.

Capoo invented an interesting language named Push-Pop. This language is an interpreted language. The interpreter starts with an empty stack with infinite capacity and reads the first instruction of the custom program. There are only two kinds of instructions in this language:

- `POP a GOTO x; PUSH b GOTO y`

  If the top element of the stack is a, then pop the stack once and transfer the control flow to the x-th instruction (which means the next instruction will be the x-th).
  Otherwise, push an element b into the stack and transfer the control flow to the y-th instruction.

- `HALT; PUSH b GOTO y`

  If the stack is empty, halt the whole program after executing this instruction.
  Otherwise, push an element b into the stack and transfer the control flow to the y-th instruction.

Capoo wants to upgrade the naive interpreter to deal with more instructions. Given a program of **at most** 1024 **instructions**, calculate the number of steps the program would execute before halting.

Input

The first line contains an integer n ($1 \le n \le 1024$), followed by n lines containing one instruction each. It is guaranteed that $1 \le a, b \le 1024,\ 1 \le x, y \le n$ for each instruction.

Output

Print -1 if the program will never halt, or the number of instructions would execute, **modulo** 998 244 353.

Examples

standard input	standard output
1 HALT; PUSH 1 GOTO 1	1
5 POP 1 GOTO 2; PUSH 1 GOTO 2 HALT; PUSH 1 GOTO 3 POP 1 GOTO 4; PUSH 2 GOTO 4 POP 1 GOTO 2; PUSH 2 GOTO 4 HALT; PUSH 99 GOTO 4	5
1 POP 1 GOTO 1; PUSH 1 GOTO 1	-1

Note

Key differences in constraints comparing to "The Empress":

- $n \le 1024$;
- The answer may not exist (program never halts), in which case report -1;
- For every program that will halt, print the result **modulo** 998 244 353.