Problem C. Halting Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program, whether the program will finish running (i.e., halt) or continue to run forever.

Alan Turing proved in 1936 that a general algorithm to solve the halting problem cannot exist, but DreamGrid, our beloved algorithm scientist, declares that he has just found a solution to the halting problem in a specific programming language – the Dream Language!

Dream Language is a programming language consists of only 5 types of instructions. All these instructions will read from or write to a 8-bit register r, whose value is initially set to 0. We now present the 5 types of instructions in the following table. Note that we denote the current instruction as the i-th instruction.

Instruction	Description
add v	Add v to the register r. As r is a 8-bit register, this instruction actually calculates $(r+v) \mod 256$ and stores the result into r, i.e. $r \leftarrow (r+v) \mod 256$. After that, go on to the $(i+1)$-th instruction.
beq v k	If the value of r is equal to v, jump to the k-th instruction, otherwise go on to the $(i+1)$-th instruction.
bne v k	If the value of r isn't equal to v, jump to the k-th instruction, otherwise go on to the $(i+1)$-th instruction.
blt v k	If the value of r is strictly smaller than v, jump to the k-th instruction, otherwise go on to the $(i+1)$-th instruction.
bgt v k	If the value of r is strictly larger than v, jump to the k-th instruction, otherwise go on to the $(i+1)$-th instruction.

A Dream Language program consisting of n instructions will always start executing from the 1st instruction, and will only halt (that is to say, stop executing) when the program tries to go on to the $(n+1)$-th instruction.

As DreamGrid's assistant, in order to help him win the Turing Award, you are asked to write a program to determine whether a given Dream Language program will eventually halt or not.

Input

There are multiple test cases. The first line of the input is an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^4$), indicating the number of instructions in the following Dream Language program.

For the following n lines, the i-th line first contains a string s ($s \in$ {"add","beq","bne","blt","bgt"}), indicating the type of the i-th instruction of the program.

- If s equals to "add", an integer v follows ($0 \le v \le 255$), indicating the value added to the register;
- Otherwise, two integers v and k follow ($0 \le v \le 255$, $1 \le k \le n$), indicating the condition value and the destination of the jump.

It's guaranteed that the sum of n of all test cases will not exceed 10^5.

Output

For each test case output one line. If the program will eventually halt, output "Yes" (without quotes); If the program will continue to run forever, output "No" (without quotes).

Example

standard input	standard output
4 2 add 1 blt 5 1 3 add 252 add 1 bgt 252 2 2 add 2 bne 7 1 3 add 1 bne 252 1 beq 252 1	Yes Yes No No

Note

For the second sample test case, note that r is a 8-bit register, so after four "add 1" instructions the value of r will change from 252 to 0, and the program will halt.

For the third sample test case, it's easy to discover that the value of r will always be even, so it's impossible for the value of r to be equal to 7, and the program will run forever.