Task 2: VaccineStorage

The COVID-19 community stores vaccines in single-column storage through an automated system. Each row of this unique storage system can store up to **1000** vaccines, and the column storage can scale up indefinitely. As vaccines are added into the single-column storage, only the top group of vaccines can be taken due to the nature of this unique storage system.

Various operations will be conducted to add, remove or move the vaccines in the storage system. The detailed list of commands is as follows:

Inject: Removes the first group of vaccines. For example, if the column has 3 groups of vaccines, each containing 4, 5 and 6 vaccines, respectively. The topmost row of vaccines, 4, would be removed. The command will fail if there are no groups left in storage.

Freeze: Adds in a group containing V number of vaccines. For example, if the column has 3 groups of vaccines, each with 4, 5 and 6 vaccines, respectively. When the command Freeze 8 is issued, the column would now look like 8, 4, 5, 6, respectively.

Combine: Compresses the top two groups of vaccines into one row. For example, if the column has 3 groups of vaccines, each with 4, 5 and 6 vaccines, respectively and the user issues this command, the column would now look like 9, 6. The command will fail if less than 2 groups are in the system or the resulting row runs out of space.

Finish: This command is issued to stop the system from taking in more inputs. Any subsequent commands will be ignored.

Your job is to process N number of commands and output the number of vaccines in the topmost row of the special column storage at the end of all operations. If any command were to fail, output `-1`. If the storage is empty, output `0`.

Input format

Your program must read from standard input.

The first line contains an input, N, which indicates the number of instructions. The following N lines contain each user command, C_i.

Output format

Your program must print to standard output.

The output consists of an integer which is the result of the first Finish instruction encountered or when a command fails.

Constraints

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances that satisfy the following limits:

- $1 \leq \boldsymbol{N} \leq 10^6$
- $\boldsymbol{C}_i$ is each one of any 4 possible commands: Inject, Freeze, Combine, Finish
- $1 \leq \boldsymbol{V} \leq 100$

Subtask	Marks	Additional Limits
1	10	$\boldsymbol{C}_i$ is only either Freeze or Finish command
2	30	$1 \leq \boldsymbol{N} \leq 10^3$
3	60	No further restrictions

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output

`4`
`Freeze 10`
`Freeze 20`
`Freeze 30`
`Finish`
	`30`

Explanation

Storage after command 1 → 10
Storage after command 2 → 20, 10

Storage after command 3 → 30, 20, 10
Finish instruction encountered at command 4.

Sample Testcase 2

This testcase is valid for subtasks 2 and 3.

Input	Output
6 Freeze 6 Freeze 4 Inject Freeze 2 Combine Finish	8

Explanation

Storage after command 1 → 6
Storage after command 2 → 4, 6
Storage after command 3 → 6
Storage after command 4 → 2, 6
Storage after command 5 → 8
Finish instruction encountered at command 6.

Sample Testcase 3

This testcase is valid for subtasks 2 and 3.

Input	Output
6 Freeze 34 Freeze 27 Inject Inject Inject Finish	-1

Explanation

Storage after command 1 → 34
Storage after command 2 → 27, 34
Storage after command 3 → 34
Storage after command 4 →
Command 5 fails as there are no vaccines left to inject.