

# **Stack Kuy**

Data Structure IUP 2022

TA's Data Structure IUP 2022

7 June 2022

## **Description**

TCenayang is a very busy student. when he was asked by his colleague (Moca) to make a program he never did. Moca is very ambitious because if this program is successful he will get an internship at an Indonesian provider company. Once upon a time Moca was offered by Germany student to complete a program called Stack Kuy. After he read and understood the plot of Stack Kuy he remembered that the program he was going to make had the same logic as Stack Kuy. The logic is:

*"First command is **"hinzufugen X Y"** to adding the value to the stack and **"subtrahieren Y"** to deleted value in stack. There are two additional commands **"ger X"** and **"gre X"**. The **"ger X"** command means that you need to add X to every integer that is in the stack, while the **"gre X"** command means that you need to subtract X from every integer that is in the stack.*

*To test the correctness of your stack operations, you need to print something like the following. Each time you perform the **"subtrahieren Y"** command, you need to print the first integer you pop. Each time you perform the **"hinzufugen X Y"** command, you need to print the size of stack.*

## **Constraint**

$0 \leq X \ \&\& \ Y \leq 4000$

$0 \leq N \leq 10$

## **Input Format**

The first line contains an integer N. The next N lines each contain an **hinzufugen X Y**, **subtrahieren Y**, **ger X**, or **gre X** command as described above. The values for X and Y are between 1 to 20.

## **Output Format**

How many lines, according to the number of "hinzufugen X Y" and "subtrahieren Y" commands given in the input. Each row contains an integer as described above.

***Example Output:***

6

hinzufugen 4 3 : [4 4 4] print —> 3

hinzufugen 5 2 : [4 4 4 5 5] print —> 5

gre 1 : [3 3 3 4 4]

ger 3 : [6 6 6 7 7]

subtrahieren 3 : [6 6] print —>7

gre 5 : [1 1]

***Then the output become:***

3

5

7