Median Updates

The median of M numbers is defined as the middle number after sorting them in order if M is odd. Or it is the average of the middle two numbers if M is even. You start with an empty number list. Then, you can add numbers to the list, or remove existing numbers from it. After each add or remove operation, output the median.

Example:
For a set of $M = 5$ numbers $9, 2, 8, 4, 1$ the median is the third number in the sorted set $1, 2, 4, 8, 9$, which is 4. Similarly, for a set of $M = 4$ numbers, $5, 2, 10, 4$, the median is the average of the second and the third element in the sorted set $2, 4, 5, 10$, which is $(4 + 5)/2 = 4.5$.

Input:
The first line is an integer, N, that indicates the number of operations. Each of the next N lines is either *a x* or *r x*. *a x* indicates that x is added to the set, and *r x* indicates that x is removed from the set.

Output:
For each operation: If the operation is *add*, output the median after adding x in a single line. If the operation is *remove* and the number x is not in the list, output *Wrong!* in a single line. If the operation is *remove* and the number x is in the list, output the median after deleting x in a single line. (If the result is an integer DO NOT output decimal point. And if the result is a real number, DO NOT output trailing 0s.)

Note
If your median is 3.0, print only 3. And if your median is 3.50, print only 3.5. Whenever you need to print the median and the list is empty, print *Wrong!*

Constraints:
$0 < N \leq 10^5$
For each *a x* or *r x*, x will always be a signed integer (which will fit in 32 bits).

Sample Input:

```
7
r 1
a 1
a 2
a 1
r 1
r 2
r 1
```

Sample Output:

```
Wrong!
1
1.5
1
1.5
1
Wrong!
```

Note: As evident from the last line of the input, if after remove operation the list becomes empty, you have to print *Wrong!*.