Pop Cat's talking

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

Pop Cat was talking in its sleep: ()()()()()(()()())()(())()((())()()())......

Pop Cat's talking in sleep can represent as a string, which only contains two character: '(' and ')'.

Your task is , for each string, to find out whether it is valid.

A string is valid if open brackets is closed in the correct order.

Input

The first line contains an integer n, which is the number of strings.

For the next n lines, each line contains a string.

Output

For each string, if it is valid print **"YE5"** (Without quotes), otherwise print **"N0"** (Without quotes).

Sample Test Data

```
4
()
(()())
)(
((())
<|==|>
YE5
YE5
N0
N0
```

Tips

Data Limit

For 50% cases: $1 <= length(S) <= 2000;$

For 100% cases: $1 <= n <= 10, 1 <= length(S) <= 100000.$

Attention

If your program computes the sample data correctly, you will get 10 points at least.

Peter's Arithmetic expression

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	4	Tradition

Description

Peter said a problem in his course. If you solve it, maybe you'll find one that looks exactly like this on the midterm.

The problem is as follows:

Gives you a infix expression, you should change it to a postfix expression.

The expression will contains only lowercase letters and arithmetic operations.

Multiplication(*) and division(/) take precedence over addition(+) and subtraction(-).

Note that there will be brackets "()" in the infix expression.

Input

A string, representing the infix expression.

Output

A string, representing the postfix expression.

The postfix expression must not contain brackets.

Sample Test Data

```
a+(b+c)*d
<|==|>
abc+d*+
```

```
a+b*c/d
<|==|>
abc*d/+
```

Tips

Data Limit

For 50% cases, it is guaranteed that the infix expression only contains addition(+), but not brackets.

For 100% cases: $1 <= length(S) <= 100$.

Attention

Asuka's queueue

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

Asuka has a queue. she call it "queueue" because it can store more elements than normal queue.

"Queueue" supports the following operations:

1. push in x of the same numbers q at the end of the queue;
2. pop out y numbers at the beginning of the queue, and print the sum of these numbers.

Initially, the queue is empty.

If the elements in the queue is less than y when doing operation 2, Asuka will likes to pop out all of them.

Please help Asuka to build this "queueue".

Input

The first line contains an integer n, which is the number of operations.

For the next n lines, each line contains several integers. **1 x q** means operation 1 and **2 y** means operation 2.

Output

For each operation 2, print the sum.

Sample Test Data

```
5
1 3 2
2 1
1 100 1
2 101
<|==|>
2
103
```

Tips

after **"1 3 2"**, the elements are 2 2 2.

after **"2 1"**, the elements are 2 2.

after **"1 100 1"**, the elements are 2 2 1 1 ... 1(100 of 1s).

after **"2 101"**, the elements are 1.

Data Limit

For 50% cases, it is guaranteed that the number of all elements do not exceed 10^6.

For 100% cases: $1 <= n <= 100000, 1 <= x <= 10^9, 1 <= q <= 10^4, 1 <= y <= 10^{18}$.

Attention

Asuka's deque

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

Asuka has a double ended queue(deque). Initially there is an element 1 in the queue. Asuka has also a global variable $c = 1$.

Asuka implements four basic operations for this deque:

1. $c = c + 1$, then push a element which equals to c to the right end of the queue;
2. pop the element at the right end of the queue;
3. $c = c + 1$, then push a element which equals to c to the left end of the queue;
4. pop the element at the left end of the queue;

However, Asuka wants to know what is in the deque. So she wants to do a special operation:

5. print the element at the middle of the queue. Formally, if the size of queue is s, print the $\frac{s+1}{2} - th$ element from the left.

But Asuka don't know how to implements the fifth operation. Please help her to finish it.

It is guaranteed that, at any time there is at least one element in the queue.

Input

The first line contains an integer n, which is the number of operations.

For the next n lines, each line contains an integer, which is the operation.

Output

For each operation 5, print the element.

Sample Test Data

```
12
1
1
5
2
5
4
5
3
3
5
4
5
<|==|>
2
1
2
4
4
```

Tips

The elements of the queue before each operation 5 are as follow:

1 2 3 → 1 2 → 2 → 5 4 2 → 4 2

Data Limit

For 50% cases, $1 <= n <= 1000$.

For 100% cases: $1 <= n <= 200000$.

Attention

Asuka's concert

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

n people are lining up to get into Asuka's concert.

People get bored waiting, so they start wandering around looking for someone they know in the line.

Any two people in a queue, A and B, can see each other if they are next to each other or if no one between them is taller than A or B. Asuka wants to know how many pairs of people can see each other.

Input

The first line contains an integer n, which is the number of operations.

For the next n lines, each line contains an integer h_i, which is the height of each person.

Output

An integer s, representing how many pairs of people can see each other.

Sample Test Data

```
7
2
4
1
2
2
5
1
<|==|>
10
```

Tips

Data Limit

For 50% cases, $1 <= n <= 1000$.

For 100% cases: $1 <= n <= 200000, 1 <= h_i <= 10^9$.

Attention