1 - Happy Jump

Information

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

Description

Given an array of integers with n elements, if the elements are all between $[1, n]$, it is called a "Happy jump". e.g. array 1423 is "Happy jump".

Given an array, your task is to decide if the array is "Happy jump" or not.

Input

The first line contains one integer n, representing the length of array.

The second line contains n integers a_i, representing the array.

Output

If the array is "Happy jump", please print "yes" (without quotes);

otherwise please print "no" (without quotes).

Sample Test Data

```
4
1 4 2 3
<|==|>
yes
```

Tips

Maybe you can use loop statements to help solving this problem.

Data Limit

For 30% cases: $1 <= n, a_i <= 10$.

For 50% cases: $1 <= n, a_i <= 100$.

For 100% cases: $1 <= n <= 10000,\ 1 <= a_i <= 10000$.

---

2 - Circular Queue

Information

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

Description

In this problem, we need to implement a circular queue.

A circular queue has its size N, a $head$ pointer which points to the first element of the queue, and a $rear$ pointer which points to the **next position** of the last element of the queue. As an initializer, $head = rear = 0$ which represents that the queue is empty. Suppose the queue stores the element at the vector with index $0, 1, 2, \ldots, N-1$.

Note that if the last element is in index $N-1$, then $rear$ will be 0, not N, because of the property of the circular queue.

Your task is to complete three functions. In each call, we use a number $k \in \{1, 2, 3\}$ to represent these functions:

$k = 1$: The **enqueue** function requires to push an integer to the rear of queue. After that, print out the index to which the $rear$ pointer points. If the queue is full (already stored N elements), print out the warning message instead.

$k = 2$: The **dequeue** function requires to pop out the first element of the queue. After that, print out the index to which the $head$ pointer points and the value of the popped element. If the queue is empty, print out the warning message instead.

$k = 3$: The **length** function prints out how many elements are in the queue.

To evaluate your task, we will call the functions q times.

Input

The first line contains two integers $N,\ q$.

For the next q lines, each line will first contains an integer k, representing the function we call. If $k = 1$, the line also contains an integer a, representing the number to push.

Output

For each **enqueue** function, print an integer representing the $rear$ index. If the queue is full, print "Queue is full"(without quotes).

For each **dequeue** function, print two integers representing the $head$ index and the popped element. If the queue is empty, print "Queue is empty"(without quotes).

For each **length** function, print an integer representing the length.

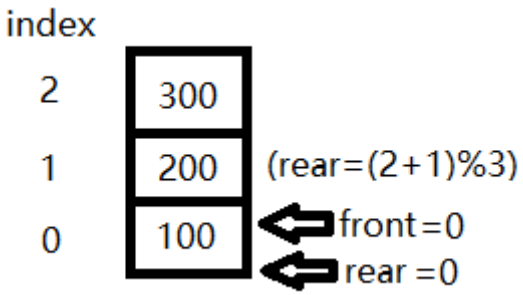
Sample Test Data

```
3 5
1 100
1 200
1 300
2
3
<|==|>
1
2
0
1 100
2
```

```
3 10
1 4551
2
2
2
1 3848
1 5543
1 8783
1 9320
2
3
<|==|>
1
1 4551
Queue is empty
Queue is empty
2
0
1
Queue is full
2 3848
2
```

Tips

In sample 1, after the first three functions are called, the queue looks like this:



Data Limit

For 30% cases: $1 <= n, q <= 10$.

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

For 100% cases: $1 <= n, q <= 10^5,\ 1 <= a <= 10000,\ k \in \{1, 2, 3\}$.

---

3 - Postfix expression

Information

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

Description

A postfix expression is an expression in which parentheses are removed, the operators are placed after two operands, and all computations are performed strictly from left to right (regardless of the precedence of the operators).

e.g. A postfix expression of $3 * (5–2) + 7$ is: $352 - *7 + @$.

Note that the integers are all between $[1, 9]$, and "@" represents the end of the whole expression.

Input

A string, representing the postfix expression.

Output

An integer, representing the answer.

Sample Test Data

```
312*-6+2/@
<|==|>
3
```

Tips

The original expression of the sample is $(3 - (1 * 2) + 6)/2$.

Note that the devision "/" is synonymous with the integer devision in Java & c++.

Data Limit

For 30% cases: $1 <= length <= 10$.

For 50% cases: $1 <= length <= 10^3$.

For 100% cases: $1 <= length <= 10^5$. guarantee that all results will not exceed the range of **int** type while computing.
