Min Queue

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Tulkip, Harul, Devu and Rushat are standing in a queue outside the vaccination centre. Since they have been waiting for a long time they need to know some information about the people in the queue.

The Nurse, Avi is busy and is not able to answer their queries and need your help in **constant time** -

- **push(x)** - enqueues the integer x
- **pop()** - dequeues the element. If the queue is empty it returns -1.
- **top()** - returns the element at the front of the queue. If the queue is empty it returns -1.
- **getMin()** - returns the minimum element in the queue. If the queue is empty it returns -1.

Note: You do not need to solve all queries in O(1) in the worst case. Do not forget to use Fast I/O for Java.

Input

The first line of input has a single integer **Q** indicating the number of queries. The following Q queries will have the given format -

- 1 x - Enqueue integer x
- 2 - Dequeue the element from the queue.
- 3 - Print the front element from the queue.
- 4 - Print the minimum element in the queue.

$1 \leq \mathbf{Q} \leq 5*10^5$, $-2^{31} \leq \mathbf{x} \leq 2^{31} - 1$

Output

For each instance of the queries 2, 3, and 4 print the answer in a new line. Print -1 in case the queue is empty(in queries of type 2,3 and 4).

Example

standard input	standard output
9 1 33 1 0 1 100 1 -10 3 4 2 2 4	33 -10 33 0 -10

Note

In the above test case -

- 33 is added to the queue
- 0 is added to the queue
- 100 is added to the queue.
- -10 is added to the queue.
- 33 is at the front of the queue and is printed.
- -10 is the minimum element in the queue and is printed.
- 33 is at the front of the queue and is printed and popped out.
- 0 is at the front of the queue and is printed and popped out.
- -10 is the minimum number in the queue and and is printed.