Task «Binary search tree»

Implement a balanced binary search tree.

You can use a built-in data structure such as std::set for this problem, but we recommend that you write your own implementation of a balanced binary search tree for practice purposes. The code you wrote will be useful in subsequent tasks.

Input format

The input to your program is a description of tree operations, their number does not exceed $100'000$. Each line contains one of the following operations:

- «insert x» – add key x to the tree. If the key x is already in the tree, then nothing needs to be done;
- «delete x» – delete key x from the tree. If the x key is not in the tree, then nothing needs to be done;
- «exists x» – if the key x exists in the tree, print «true», otherwise «false»;
- «next x» – print the minimum element in the tree that is strictly greater than x, or «none» if there is none;
- «prev x» – print the maximum element in the tree strictly less than x, or «none» if there is none.

All numbers in the input file are integers and modulo does not exceed 10^9.

Output format

Output sequentially the result of all operations exists, next, prev. Print the result of each operation on a separate line.

Sample input:

```
insert 2
insert 5
insert 3
exists 2
exists 4
next 4
prev 4
delete 5
next 4
prev 4
```

Sample output:

```
true
false
5
3
none
3
```