BstTree

Youre task is to implement **Binary Search Tree** data structure: `bstTree`. Your implementation should support the following operations:

- `INSERT X`, which inserts key x to `bstTree`,
- `SEARCH X`, which searches for key x in `bstTree`,
- `PREORDER`, which outputs the keys of `bstTree` sorted by **preorder** rule,
- `INORDER`, which outputs the keys of `bstTree` sorted by **inorder** rule,
- `POSTORDER`, which outputs the keys of `bstTree` sorted by **postorder** rule.

Input

The first line contains an integer z ($1 \leqslant z \leqslant 2 \cdot 10^9$) – the number of data sets. Each data set is as follows:

The first line contains a number n ($1 \leqslant n \leqslant 4000000$) – the number of the operations performed on bstTree. Each of the next n lines contains an instruction (with an argument if applied) to be performed on `bstTree`.

Output

Each instruction should produce the following output:

- `INSERT x` outputs 1 if x is added succesfully to `bstTree`, 0 otherwise;
- `SEARCH X` outputs 1 if x is in `bstTree`, 0 otherwise;
- `PREORDER, INORDER, POSTORDER` output the keys of `bstTree` sorted by **preorder, inorder, postorder** rule, respectively.

Example

For the input:

```
1
17
INSERT 7
INSERT 4
INSERT 2
INSERT 5
INSERT 12
INSERT 11
INSERT 9
INSERT 8
INSERT 10
INSERT 13
INORDER
POSTORDER
PREORDER
SEARCH 1
SEARCH 15
SEARCH 9
INSERT 7
```

the output is:

```
1
1
1
1
1
1
1
1
1
2 4 5 7 8 9 10 11 12 13
2 5 4 8 10 9 11 13 12 7
7 4 2 5 12 11 9 8 10 13
0
0
1
0
```