

# Very Simple BST

Implementasi Lah *Binary Search Tree* (BST) sederhana (tidak seimbang) yang menyimpan bilangan bulat dan mendukung 4 jenis operasi:

- **INSERT x:** Masukkan kunci x ke dalam BST. Jika x sudah ada di dalam pohon, abaikan operasi ini.
- **DELETE x:** Hapus kunci x dari BST. Jika x tidak ada di dalam pohon, abaikan operasi ini.
- **SEARCH x:** Cetak kedalaman dari node x. Kedalaman root adalah 1. Jika x tidak ditemukan, cetak -1.
- **PRINT:** Cetak semua kunci di dalam BST secara in-order traversal, dipisahkan oleh spasi. Jika pohon kosong, cetak EMPTY.

## Input

Baris pertama berisi sebuah bilangan bulat N, yaitu jumlah total operasi. N baris berikutnya masing-masing berisi satu perintah operasi.

## Output

Untuk setiap perintah SEARCH atau PRINT, cetak hasilnya pada baris baru sesuai dengan format yang telah ditentukan.

## CONSTRAINTS

$$1 \leq N \leq 1000$$

Sample input 1:

```
11
INSERT 20
INSERT 10
INSERT 30
INSERT 5
INSERT 15
SEARCH 15
PRINT
DELETE 20
SEARCH 10
PRINT
SEARCH 99
```

Sample output 1:

```
3
5 10 15 20 30
```

2  
5 10 15 30  
-1

Sample input 2:

7  
PRINT  
INSERT 50  
INSERT 50  
PRINT  
DELETE 10  
DELETE 50  
PRINT

Sample output 2:

EMPTY  
50  
EMPTY

# Very Simple BST

Implement a plain (unbalanced) Binary Search Tree (BST) of integers that supports four types of operations.

- **INSERT x:** Insert the key x into the BST. If x already exists in the tree, do nothing.
- **DELETE x:** Remove the key x from the BST. If x is not present in the tree, do nothing.
- **SEARCH x:** Print the depth of node x. The root's depth is 1. If x is not found, print -1.
- **PRINT:** Print all keys in the BST using an in-order traversal, separated by spaces. If the tree is empty, print EMPTY.

## Input

The first line contains a single integer N, the total number of operations. The next N lines each contain one operation command.

## Output

For each SEARCH or PRINT command, print the result on a new line according to the specified format.

## CONSTRAINTS

$$1 \leq N \leq 1000$$

Sample input 1:

```
11
INSERT 20
INSERT 10
INSERT 30
INSERT 5
INSERT 15
SEARCH 15
PRINT
DELETE 20
SEARCH 10
PRINT
SEARCH 99
```

Sample output 1:

```
3
5 10 15 20 30
2
```

5 10 15 30  
-1

Sample input 2:

7  
PRINT  
INSERT 50  
INSERT 50  
PRINT  
DELETE 10  
DELETE 50  
PRINT

Sample output 2:

EMPTY  
50  
EMPTY