

Element Present in a Tree

[View](#)[Try](#)[Insights](#)

🏆 Points: 50 ✅ 14 test cases (1 sample)

[Coding](#)[Data Structures](#)[Binary Search Tree](#)[EASY](#)[Trees](#)[Algorithms](#)[Core Skills](#)[Problem Solving](#)

Each node of a *Binary Search Tree (BST)* has an integer value and pointers to as many as two children. The children of a parent node are referred to as the *left child* and the *right child*. The value of *left child* is always less than the value of its parent node, and the value of *right child* is always greater than or equal to the value of its parent node.

Function Description

Complete the function *isPresent* in the editor below. The function must return *1* if the value is present in the BST, or *0* if it's not.

isPresent has the following parameter(s):

root: reference to the root node of a singly-linked list of integers

val: integer to search for

Constraints

- $1 \leq \text{total nodes} \leq 10^5$
- $1 \leq \text{val} \leq 5 \times 10^4$

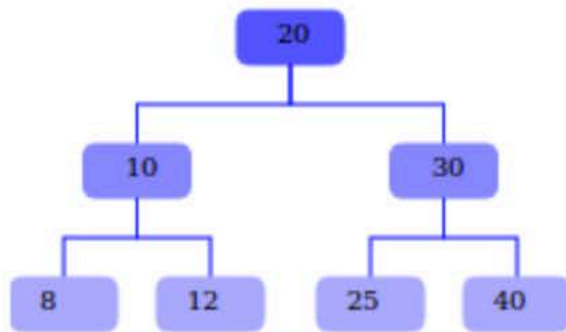
▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function:

Each line contains an integer to search for in the given BST.

▼ Sample Case 0

Sample Input 0



Values

```
30
10
12
15
```

Sample Output 0

```
1
1
```

Sample Output 0

```
1
1
1
0
```

Explanation

Value: 30. This value is *present* in the *BST*, so *isPresent* returns 1.

Value: 10. This value is *present* in the *BST*, so *isPresent* returns 1.

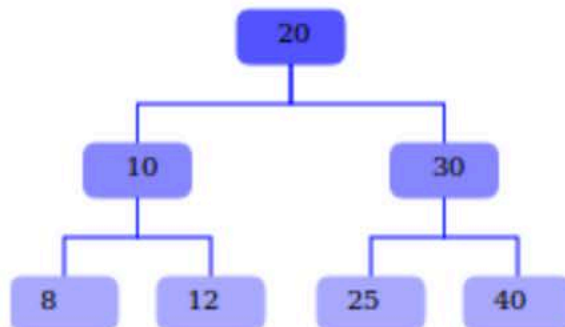
Value: 12. This value is *present* in the *BST*, so *isPresent* returns 1.

Value: 15. This value is *not present* in the *BST*, so *isPresent* returns 0.

▼ Sample Case 1

Sample Input 1

Click bar to open/close the example.



Values

```
79
10
20
30
40
```

Sample Output 1

```
0
1
1
1
1
```

Click bar to open/close the example.

Explanation

Value: 79. This value is *not present* in the *BST*, so *isPresent* returns 0.

Value: 10. This value is *present* in the *BST*, so *isPresent* returns 1.

Value: 20. This value is *present* in the *BST*, so *isPresent* returns 1.

Value: 30. This value is *present* in the *BST*, so *isPresent* returns 1.

Value: 40. This value is *present* in the *BST*, so *isPresent* returns 1.