

<https://course.acciojob.com/idle?question=3a25fd86-4629-49a4-ae39-abb4978e4a4d>

● MEDIUM

● Max Score: 40 Points

Search a node in BST

You are given the root of a binary search tree (BST) and an integer val.

Find the node in the BST that the node's value equals val and return true. If such a node does not exist, return false.

Input Format

The first line inputs N , the number of nodes and x , value of the node to find.

The second line inputs the value of N nodes of the BST.

Output Format

Print "YES" if node is present else "NO" in a new line.

Example 1

Input

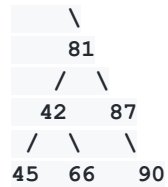
```
7 87
2 81 42 87 90 42 45 66
```

Output

YES

Explanation

2



As 87 is present in the given nodes , so the output will be YES.

Example 2

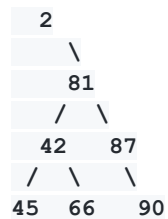
Input

```
7 69
2 81 42 87 90 42 45 66
```

Output

NO

Explanation



As 69 is not present in the given nodes , so the output will be NO.

Constraints

$1 \leq N \leq 1000$

$1000 \leq \text{val}[\text{node}] \leq 1000$

Topic Tags

- **Recursion**

- Trees
- BST

My code

```
// n java
import java.util.*;
import java.lang.*;
import java.io.*;

class Node
{
    int data;
    Node next ,prev;

    Node(int data, Node next,Node prev)
    {
        this.data = data;
        this.next = next;
        this.prev = prev;
    }

    Node() {}
}

public class Main
{
    static Node insert(Node root,int n)
    {
```

```

    if(root==null)
    {
        root=new Node(n,null,null);
        return root;
    }
    else if(n< root.data)
        root.prev= insert( root.prev, n);
    else if(n>root.data)
        root.next= insert( root.next, n);
    return root;
}
static void seartch(Node root,int n)
{
    if(root==null)
    {
        System.out.print("NO");
        return;
    }
    else if(n< root.data)
        seartch( root.prev, n);
    else if(n>root.data)
        seartch( root.next, n);
    else if(n==root.data)
    {
        System.out.print("YES");
        return;
    }
}
}

```

```
public static void main (String[] args) throws
java.lang.Exception
{
    //your code here
    Scanner s=new Scanner(System.in);
    int n=s.nextInt();
    int k=s.nextInt();
    //int arr[]=new int[n];
    Node root=null;
    for(int i=0;i<n;i++)
    {
        int m=s.nextInt();
        root=insert( root, m);
    }
    seartch(root,k);
}
}
```