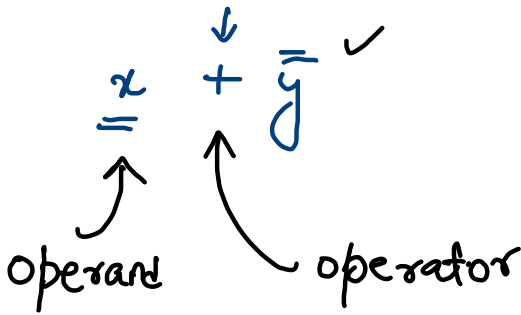
Postfix expression calculation

- Given a string **Str** in **Postfix expression** calculate the result of this expression.
- String has 2 types of char.

```
- case 1: char type1 = [0-9]
- case 2: char type2 = [+, - , / ,* ]
```

- (**Note** : It can be assumed that the expression is always valid.)



Sample Input 0

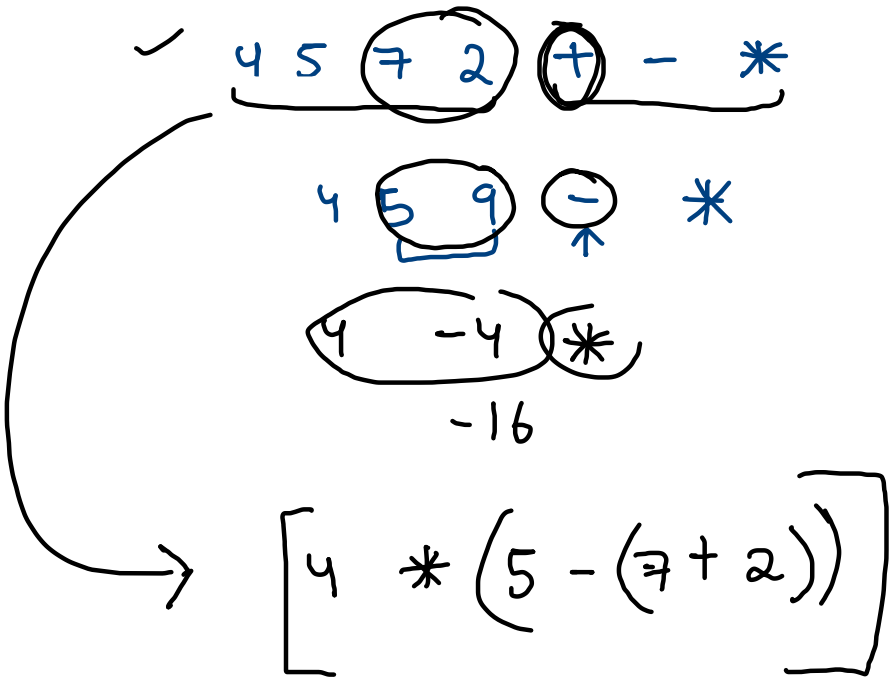
```
4572+-*
```

Sample Output 0

```
-16
```

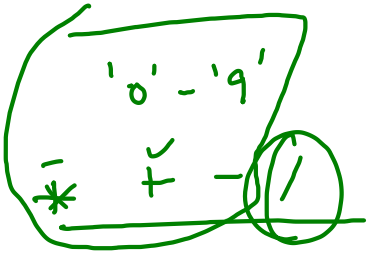
Explanation 0

Infix expression : 4 * (5 - (7 + 2)) => -16

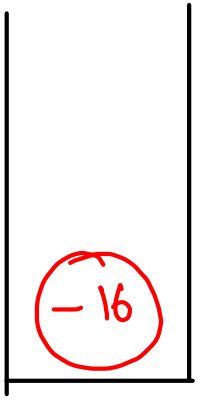


4 5 7 2

st → operand



st
Int




```java
import java.io.*;
import java.util.*;
public class Solution {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        Stack<Integer> st = new Stack<>();
        for(int i = 0; i <s.length(); i++){
            char ch = s.charAt(i);
            if(ch >= '0' && ch <= '9'){
                st.push(ch - '0');
            }
            else{ //operator
                int v2 = st.pop();
                int v1 = st.pop();
                if(ch == '+'){
                    st.push(v1+v2);
                }else if(ch == '-'){
                    st.push(v1-v2);
                }else if(ch == '*'){
                    st.push(v1*v2);
                }else{
                    st.push(v1/v2);
                }
            }
        }
        System.out.println(st.peek());
    }
}
```

Next Smaller Element To The Right

Problem | Submissions | Leaderboard | Discussions

1. You are given a number n, representing the size of array a
2. You are given n numbers, representing elements of array a.
3. You are required to "next smaller element on the right" for all elements of array
4. Input and output is handled for you.

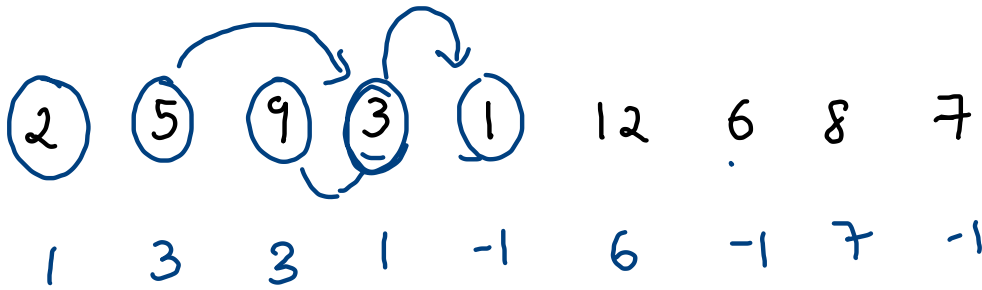
"Next smaller element on the right" of an element x is defined as the first element to right of x having value smaller than x. Note -> If an element does not have any element on it's right side smaller than it, consider -1 as it's "next smaller element on right"

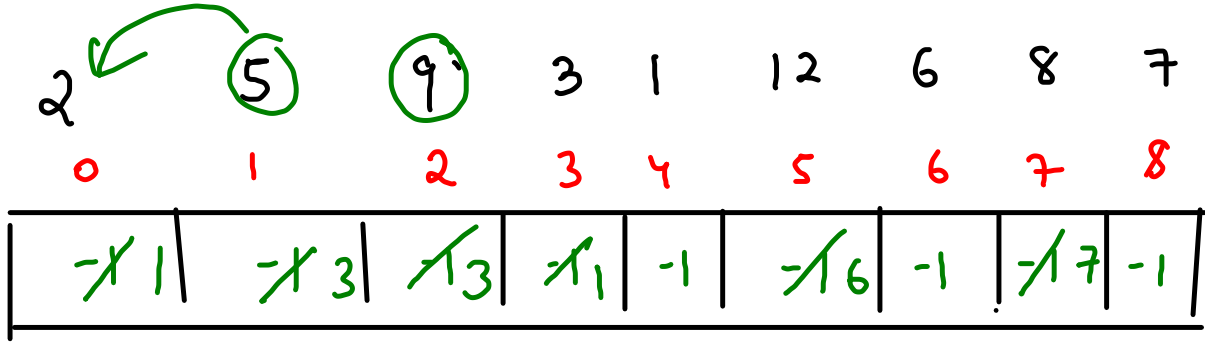
Sample Input 0

```
9
2 5 9 3 1 12 6 8 7
```

Sample Output 0

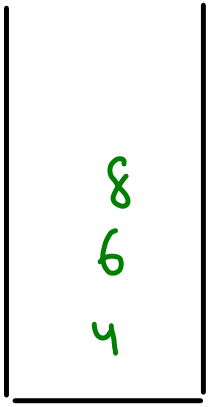
```
1 3 3 1 -1 6 -1 7 -1
```





st

idx

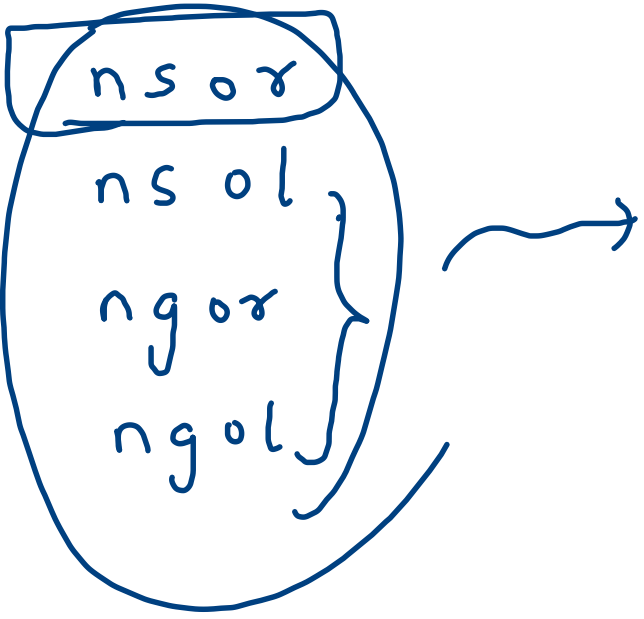


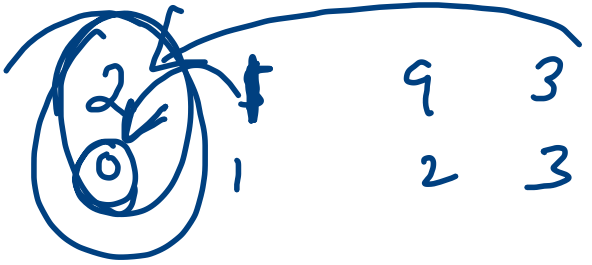
i = 9

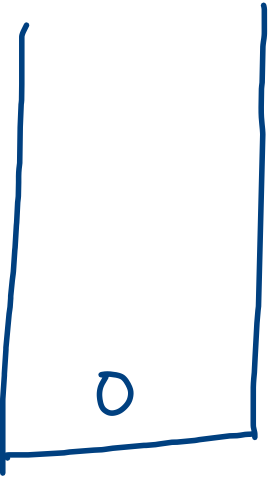
v =

while

A[i] < A[st.peek]


nsor
nsol
ngor
ngol


2
9
3
1
0
1
2
3

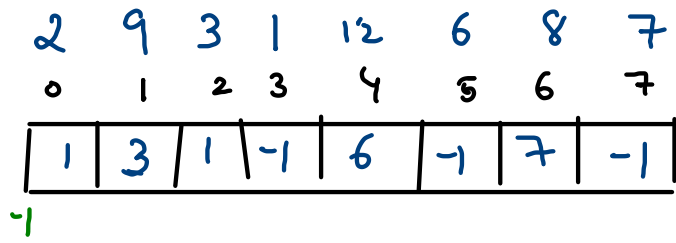

0

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        //logic
        int [] ans = new int[n];
        for(int i = 0; i < n; i++){
            ans[i] = -1;
        }                           //instead write Arrays.fill(ans, -1);
        // Arrays.fill(ans, -1);
        Stack<Integer> st = new Stack<>();
        st.push(0);
        for(int i = 1; i < n; i++){
            while(st.size() != 0 && A[i] < A[st.peek()]){
                int idx = st.pop();
                ans[idx] = A[i];
            }
            st.push(i);
        }
        for(int e : ans){
            System.out.println(e + " ");
        }
    }
}
```

Asteroid.



i=

7 < 8

7 < 6

Hashmap.

→ k, V

initialize

add

get

size

remove

* check if k is present

* getOrDefault.

Main.java

```java
import java.util.HashMap;
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        //init -- K, V
        HashMap<String , Integer> hm = new  HashMap<>();

        //add -- put: 2 Type insert or update
        hm.put("Honda", 18);
        hm.put("Suzuki", 12);
        hm.put("BMW", 8);
        hm.put("Honda", 15);
        //size
        System.out.println(hm.size());
        //get
        System.out.println(hm.get("Honda"));
        //remove
        hm.remove("BMW");
        System.out.println(hm);
    }
}
```

```java
import java.util.HashMap;
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        //init -- K, V
        HashMap<String , Integer> hm = new  HashMap<>();

        //add -- put: 2 Type insert or update
        hm.put("Honda", 18);
        hm.put("Suzuki", 12);
        hm.put("BMW", 8);
        hm.put("Honda", 15);

        hm.remove("BMW");
        System.out.println(hm.getOrDefault("BMW", 0));

    }
}

```