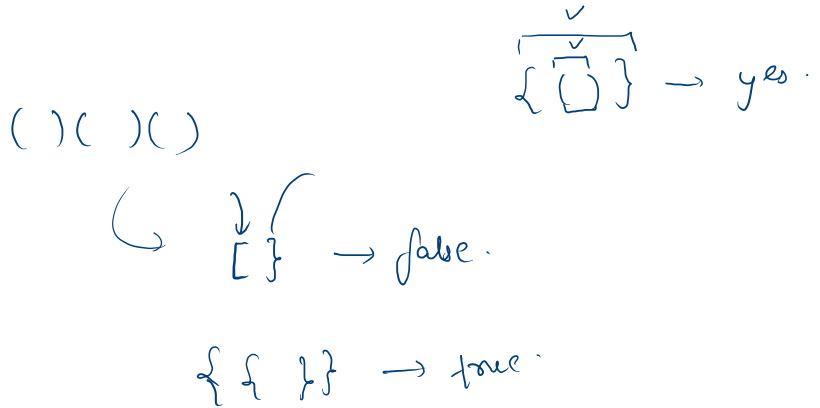
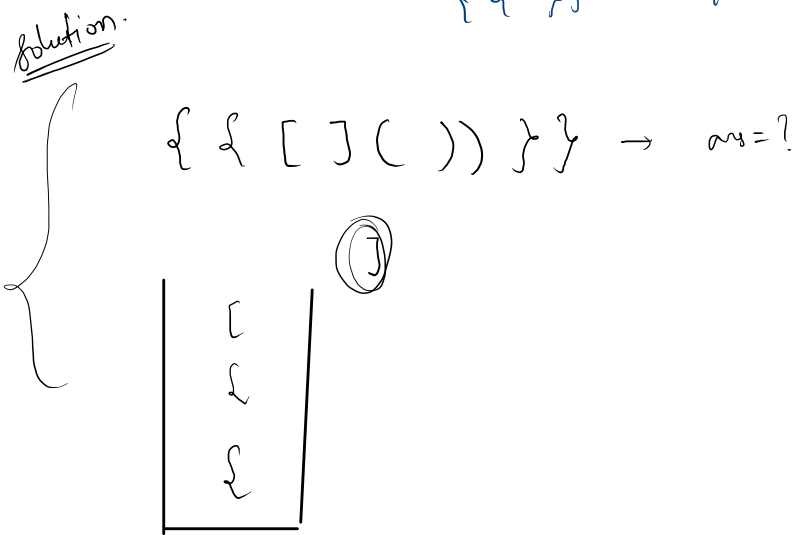
Valid Parentheses 4

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

- Open brackets must be closed by the same type of brackets.
- Open brackets must be closed in the correct order.



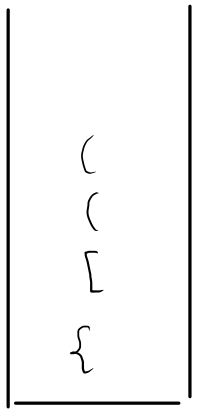


s → { [(( ] ((} → false.

0 1 2 3 4 5 6 7

i

]



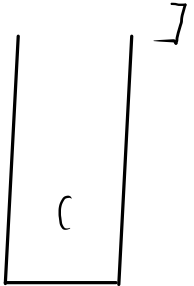
[({

]

eg

( ]

]

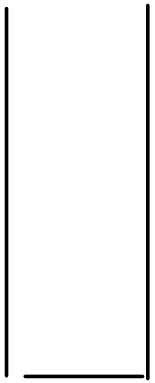


( ] → false.

s → "]...." "}...." ")...." ✓

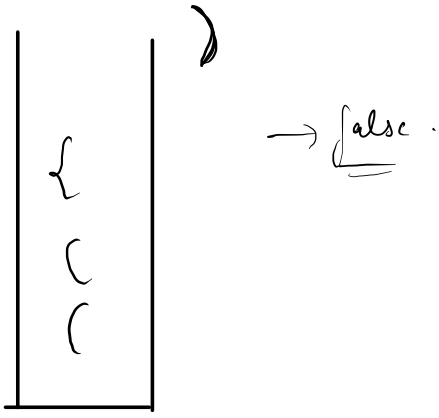
s → {()[()]} → ans = ? true
0 1 2 3 4 5 6 7

i



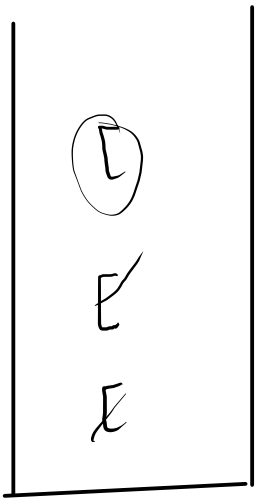
( )

s → (({) [ ]



→ false .

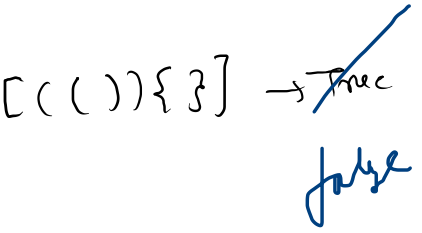
s→ [ ][ ][.



st.size != 0

↳ false.

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

public class Solution {
    public static boolean findAns(String s){
        Stack<Character> st = new Stack<>();
        for(int i = 0; i < s.length();i++){
            char ch = s.charAt(i);
            if(ch == '{' || ch == '[' || ch == '('){
                st.push(ch);
            }else if(st.size() == 0){
                return false;
            }else if(ch == '}' && st.peek() != '{'){
                return false;
            }
            else if(ch == ')' && st.peek() != '('){
                return false;
            }
            else if(ch == ']' && st.peek() != '['){
                return false;
            }else{
                st.pop();
            }
        }
        return st.size() == 0;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        boolean ans = findAns(s);
        System.out.println(ans);
    }
}
```



Hashmap.

↳ D.S.

↳ key - value.

key	value.

* key is always unique.

initialize

size

get

add

How many schools are there in different countries?

eg.

key	value
India	46
China	52
Canada	12
Nepal	70

Key / Value

String
Integer
Character
.
.
.

eg. Temperature

Delhi - 33°
Mumbai - 29°
Kolkata - 24°
Punjab - 30°

add → put

- add → if key is not present
- update → if key is present

```java
import java.util.HashMap;
import java.util.*;

public class Main
{
    public static void main(String[] args) {
        //init
        HashMap<String , Integer> hm = new HashMap<>();
        //add -> put
        hm.put("China", 38);
        hm.put("India", 42);
        hm.put("Nepal", 70);
        hm.put("Canada", 22);
        System.out.println(hm);

        hm.put("India", 55);

        System.out.println(hm);
    }
}
```

size → total unique entries / key.

```
System.out.println(hm.size());
```

get → hm.get(key)

```
18        System.out.println(hm.get("Nepal"));
```

→ 70

remove

```
17        hm.remove("Canada");
```

hm.remove(key);

hm.keySet()

```
        System.out.println(hm.keySet());
    }
}
```

all keys.

hm.containsKey()

```
System.out.println(hm.containsKey("China"));
```

hm.getOrDefault()

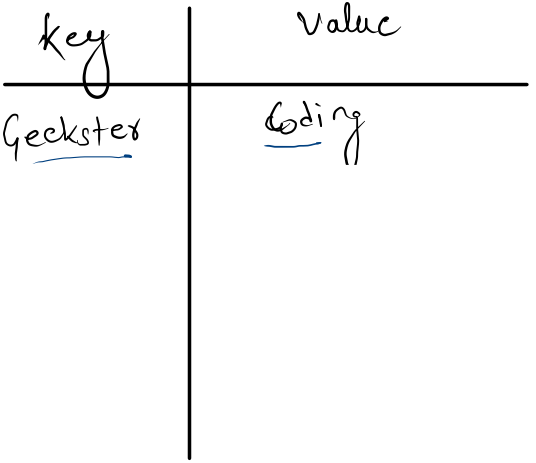
```
        String key = "abc";
        // if(hm.containsKey(key)){
        //     System.out.println(hm.get(key));
        // }else{
        //     System.out.println(-1);
        // }

        System.out.println(hm.getOrDefault(key, -1));

```

Word Meaning

- If **N==1**, take **word** and **meaning** as input from user and **add** it to the dictionary.
- If **N==2**, take a **word** as input from the user and print its **meaning**, if the word is not found print **-1**.
- If **N==3**, take a **word** as input from the user and delete it from the dictionary.
- If **N==4**, **Close** the dictionary(**Exit** the program).

Sample Input 1

```
1
Geekster
Coding
1
Geek
Coder
2
Geek
3
Geek
2
Geekster
2
Geek
4
```



Sample Output 1

```
Coder
Coding
-1
```

while (condition)

→ true

{

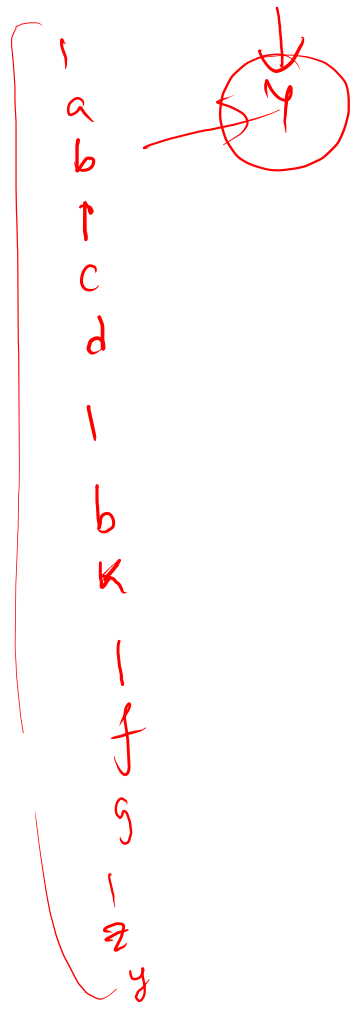
}

while (true)

{

→ stop. (break)

}



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

public class Solution {

    public static void main(String[] args) {
        HashMap<String, String> hm = new HashMap<>();
        Scanner scn = new Scanner(System.in);
        while(true){
            int n = scn.nextInt();
            if(n == 1){
                String word = scn.next();
                String meaning  = scn.next();
                hm.put(word, meaning);
            }else if(n == 2){
                String word = scn.next();
                System.out.println(hm.getOrDefault(word, "-1"));
            }else if(n == 3){
                String word = scn.next();
                hm.remove(word);

            }else if(n == 4){
                break;
            }
        }

    }
}
```