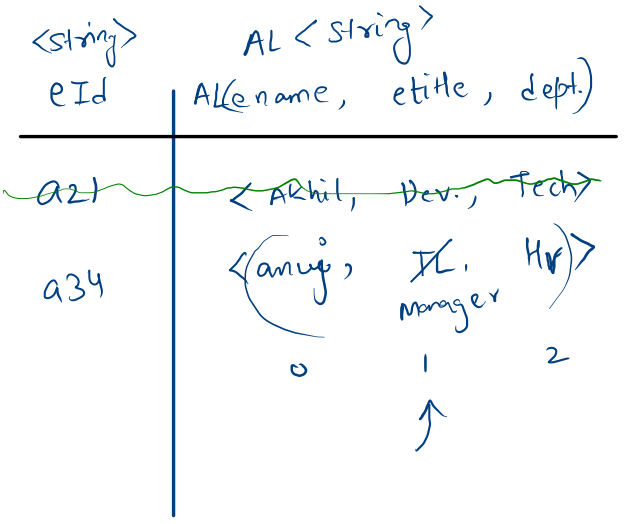
employee management

You are tasked with developing an employee management system for a company. To efficiently store employee data, you decide to use a **HashMap>**. In this HashMap, the keys represent unique employee IDs, and the values are ArrayLists of employee details as strings, including the employee's name, job title, and department.

you will be getting **T** queries which includes:

1. case-1 (add) -> add employee with details.
2. case-2 (update) -> update job title of a given employee.
3. case-3 (delete) -> remove the employee.
4. case-4 (show) -> print details of a given employee else print -1.

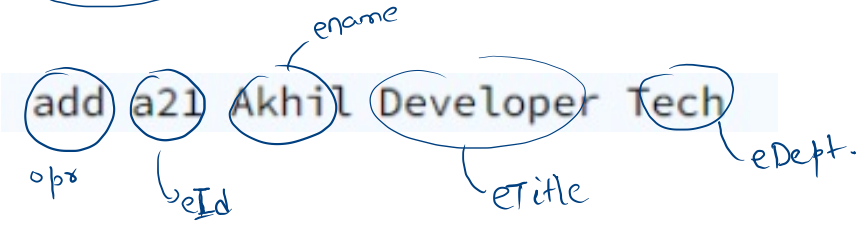


Sample Input 0

```
5
add a21 Akhil Developer Tech
add a34 anuj TeamLead Hr
update a34 Manager
delete a21
show a34
```

Sample Output 0

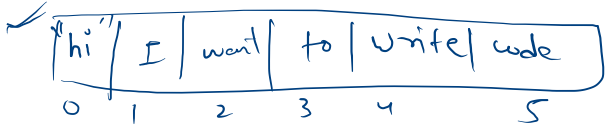
```
anuj Manager Hr
```



```java
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        HashMap<String, ArrayList<String>> hm = new HashMap<>();
        int n = scn.nextInt();
        for(int i = 0; i < n; i++){
            String opr = scn.next();
            String id = scn.next();

            if(opr.equals("add")){
                String name = scn.next();
                String title = scn.next();
                String dept = scn.next();
                ArrayList<String> details = new ArrayList<>();
                details.add(name);
                details.add(title);
                details.add(dept);
                hm.put(id, details);

            }else if(opr.equals("update")){
                String newRole = scn.next();
                ArrayList<String> data = hm.get(id);
                data.set(1, newRole);
                hm.put(id, data);
            }else if(opr.equals("delete")){
                hm.remove(id);
            }else{//show
                if(hm.containsKey(id)){
                    for(String res: hm.get(id)){
                        System.out.print(res + " ");
                    }
                    System.out.println();
                }
                else{
                     System.out.println(-1);
                }

            }
        }


    }
}
```

```java
public class Main
{
    public static void main(String[] args) {
        String s = "hi I want to write code";
        String [] A = s.split(" ");

        System.out.println(A.length);

        for(String r : A){
            System.out.println(r);
        }
        System.out.println();
    }
}
```

s → "hi I want to write code"



Two Sum 14

Sample Input 0

```
4 9
2 7 11 15
```

Sample Output 0

```
0 1
```

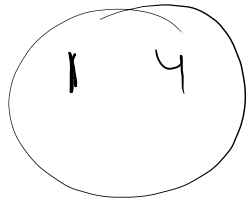
tar = 9

A[i] = ~~X~~ 2 ~~15~~ ~~11~~ 7

rem = tar - A[i] = ~~8~~ ~~A~~ ~~-6~~ ~~-2~~ 2

1	2	15	11	7
0	1	2	3	4
				i

ele	idx
1	0
2	1
15	2
11	3



```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 tar = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }

        HashMap<Integer, Integer> hm = new HashMap<>();

        for(int i = 0; i < n; i++){
            int rem = tar - A[i];
            if(hm.containsKey(rem)){
                System.out.println(hm.get(rem) + " " + i);
                break;
            }
            hm.put(A[i], i);
        }
    }
}
```

tar = 9

1	2	5	7	6
0	1	2	3	4

i

rem = 2

K	V
1	0
2	1
5	2

1 3

Max Number of K-Sum Pairs

eg. $k = 5$

1	2	3	4	6	7	1	4	3
0	1	2	3	4	5	6	7	8

i

1. freq map.

Sample Input 0

```
4
1 2 3 4
5
```

Sample Output 0

```
2
```

6	1
7	1
3	1

if (freq == 0) remove

curr = A[i] = 3

rem = k - A[i] = 2

4 + 1

ans = ~~0~~ ~~1~~ 2 3

```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();
        }
        int k = scn.nextInt();
        int ans = 0;
        HashMap<Integer, Integer> hm = new HashMap<>();
        for(int i = 0; i < n; i++){
            int rem = k - A[i];
            if(hm.containsKey(rem)){
                ans++;
                int val = hm.get(rem)-1;
                hm.put(rem, val);
                if(val == 0){
                    hm.remove(rem);
                }
            }else{
                hm.put(A[i], hm.getOrDefault(A[i], 0) + 1);
            }
        }
        System.out.println(ans);
    }
}
```

ans = 0