Phone Directory By Name

Problem | Submissions | Leaderboard | Discussions

You are required to create a **Phone Directory** consisting of Name and corresponding Phone Number.

Take an integer N as input and **Continue** the process untill **Case 4** is not achieved.

- If **N==1**, take **Name** and **Phone Number** as input from user and **add** it to the Phone Directory.
- If **N==2**, take a **Name** as input from the user and print corresponding **Phone Number**, if the **Name** is not found print **-1**.
- If **N==3**, take a **Name** as input from the user and delete it from the Phone Directory.
- If **N==4**, **Close** the Directory(**Exit** the program).

Note: Take Phone Number as String input.

Sample Input 0

```
1
Geekster
9876543210
1
Geeku
0123456789
2
Geeku
2
Geekster
3
Geeku
2
Geeku
2
Geekster
4
```

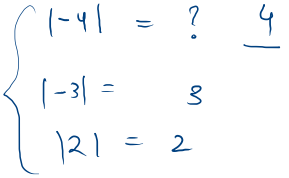
Sample Output 0

```
0123456789
9876543210
-1
9876543210
```

Main.java

```java
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        HashMap<String , Integer> hm = new HashMap<>();
        hm.put("Aman", 272);
        hm.put("Riya", 403);
        hm.put("Arnav", 72);
        hm.put("Pankaj", 670);
        hm.put("Kartik", 144);

       // System.out.println(hm);


        for(String k : hm.keySet()){
            System.out.println(k + "--" + hm.get(k) );
        }

        //Aman -- 272
        //Pankaj -- 670
    }
}

```

Same Number Same Frequency



Problem | Submissions | Leaderboard | Discussions

Take an **Integer N** as input and then take **N integers** input from **Geeku**.

Geeku wants to print all those integers whose frequency is **exactly same** as the integer's absolute value.

You have to help **Geeku** in doing so.

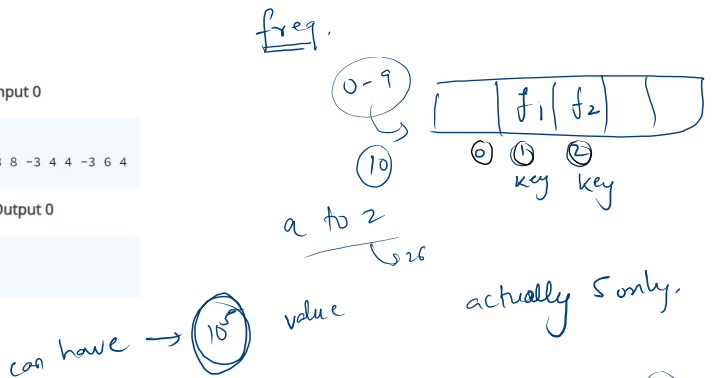
Note: 0 is excluded

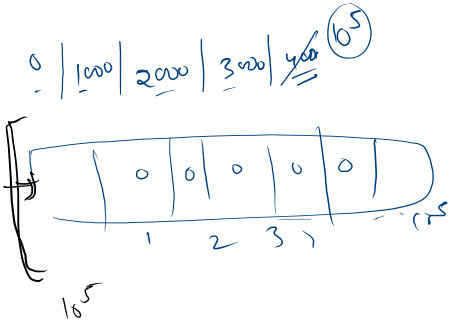
Sample Input 0

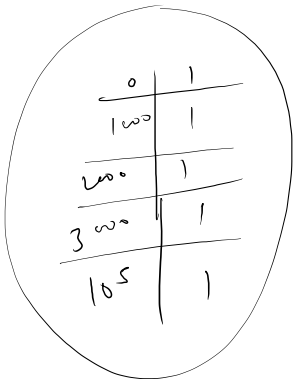
```
10
4 5 -3 8 -3 4 4 -3 6 4
```

Sample Output 0

```
-3
4
```





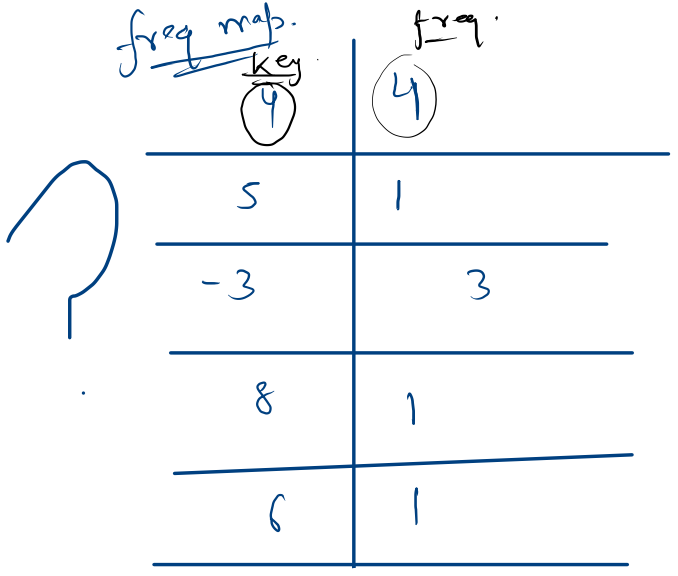


Sample Input 0

```
10
4 5 -3 8 -3 4 4 -3 6 4
```

Sample Output 0

```
-3
4
```



key	val
4	4
|-3|	3

|key| == freq
↳ key

Sample Input 0

```
10
4 5 -3 8 -3 4 4 -3 6 4
```

i

1. i/p → Array.
2. freq → map.

k	v
4	4
5	1
-3	3
8	1
6	1

if |k| == v → k

```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();
        }
        HashMap<Integer, Integer> hm = new HashMap<>();
        for(int i = 0; i < n; i++){
            if(hm.containsKey(A[i])){
                hm.put(A[i], hm.get(A[i])+1);
            }else{
                hm.put(A[i], 1);
            }
        }
        System.out.print(hm);
    }
}
```

-3 -3 -3 4 4 4 4 5 6 8

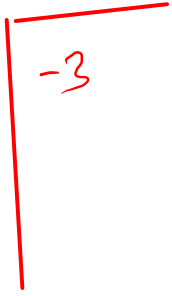
-3 -3 -3 4 4 4 5 6 8

i

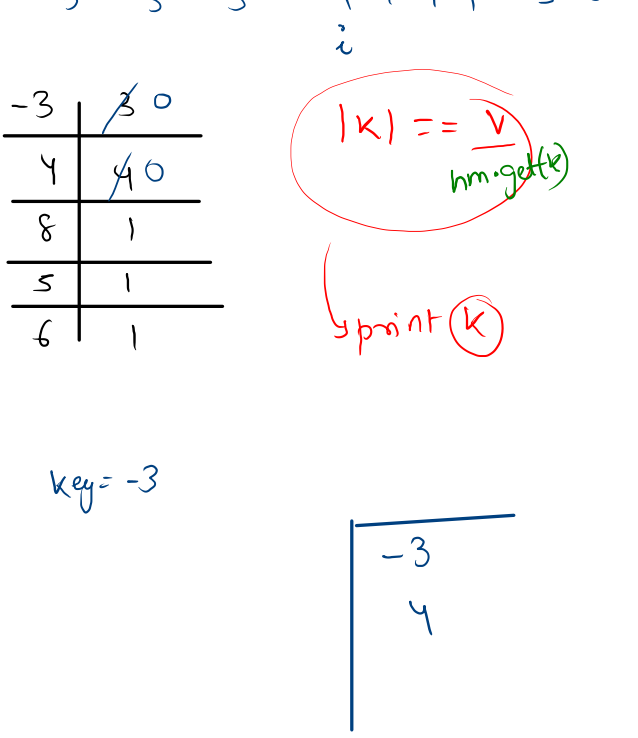
key = -3

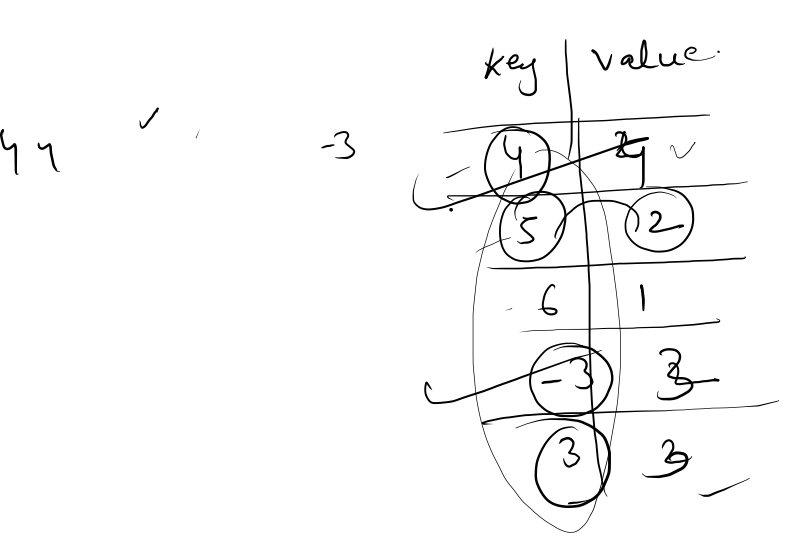
|-3| == 3

```
22
23          for(int i = 0; i < n; i++){
24              int key = A[i];
25              if(Math.abs(key) == hm.get(key)){
26                  System.out.println(key);
27              }
28          }
29
30      }
```



```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();
        }
        Arrays.sort(A); ////just to match with testcase for geekster
        HashMap<Integer, Integer> hm = new HashMap<>();
        for(int i = 0; i < n; i++){
            if(hm.containsKey(A[i])){
                hm.put(A[i], hm.get(A[i])+1);
            }else{
                hm.put(A[i], 1);
            }
        }

        for(int i = 0; i < n; i++){
            int key = A[i];
            if(Math.abs(key) == hm.get(key)){
                System.out.println(key);
                hm.put(key, 0);
            }
        }

    }
}
```



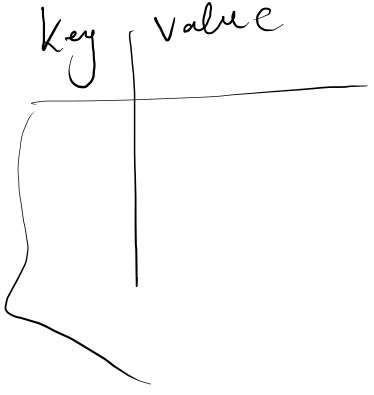


if (|key| == value)

key.

|4| == 4

|3| == 3



Character and it's Frequency

Problem | Submissions | Leaderboard | Discussions

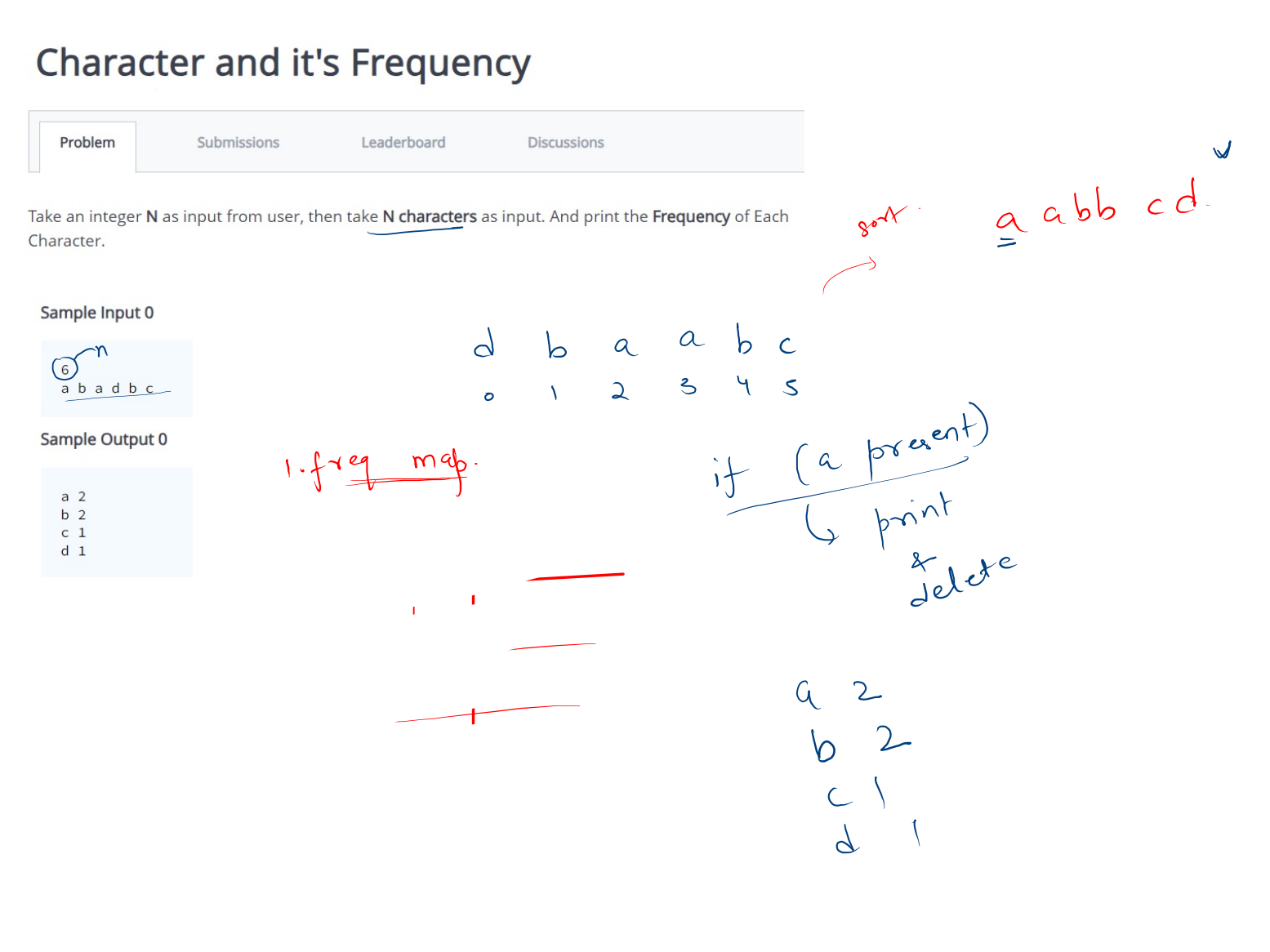
Take an integer **N** as input from user, then take **N characters** as input. And print the **Frequency** of Each Character.

Sample Input 0

```
6
a b a d b c
```

Sample Output 0

```
a 2
b 2
c 1
d 1
```



```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();
        char [] A = new char[n];
        for(int i = 0; i < n; i++){
           A[i] = scn.next().charAt(0);
        }
        Arrays.sort(A);
        HashMap<Character, Integer> hm = new HashMap<>();
        //freq map
        for(int i = 0; i < n; i++){
            if(hm.containsKey(A[i])){
                hm.put(A[i], hm.get(A[i]) + 1);
            }else{
                hm.put(A[i], 1);
            }
        }
        //
        for(int i = 0; i < n; i++){
            if(hm.containsKey(A[i])){
                //print nd remove
                System.out.println(A[i] + " " + hm.get(A[i]));
                hm.remove(A[i]);
            }
        }

    }
}
```

dry run.

d	c	c	a	a	b	b	b
0	1	2	3	4	5	6	7

employee management

Problem | Submissions | Leaderboard | Discussions

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.

str → name job title dpt.

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
```