Started on	Tuesday, 2 January 2024, 11:26 AM
State	Finished
Completed on	Tuesday, 2 January 2024, 1:01 PM
Time taken	1 hour 34 mins

Question **1**

Correct

Marked out of 25.00

Try using a Set in your application. The property of Set is, it doesn't allow duplicate elements and does not maintain order like a list. Understand it by going through and completing the problem.

Create a driver class called Main.

In the Main method, obtain username input from the user.

Add the usernames to a set.

Display the number of unique usernames at the end of the program.

Input Format :

The first line of the input consists of the number of users.

Next input is the user names.

Output Format :

The output displays the number of unique elements in the set.

Sample testcases :

Testcase 1 Input

4

Ram

Christopher

Ahamed

Ahamed

Testcase 1 Output

3

For example:

Input	Result
4 Ram Christopher Ahamed Ahamed	3

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class UniqueNames
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        HashSet<String> names = new HashSet<String>();
        int n = sc.nextInt();
        sc.nextLine();
        for(int i=0;i<n;i++)
        {
            names.add(sc.nextLine());

        }
        System.out.println(names.size());
    }
}
```

	Input	Expected	Got	
✓	4 Ram Christopher Ahamed Ahamed	3	3	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 25.00

Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct.

Example 1:

Input:

4

1 2 3 1

Output: true

Example 2:

Input:

4

1 2 3 4

Output: false

Example 3:

Input:

10

1 1 1 3 3 4 3 2 4 2

Output: true

Constraints:

1 <= nums.length <= 10^5

-109 <= nums[i] <= 10^9

For example:

Input	Result
4 1 2 3 1	true
4 1 2 3 4	false
10 1 1 1 3 3 4 3 2 4 2	true

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class RepeatedElements
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> nums = new ArrayList<Integer>();
        int n = sc.nextInt();
        for(int i=0;i<n;i++)
        {
            nums.add(sc.nextInt());
        }
        HashSet<Integer> freq = new HashSet<Integer>();
        for(int i:nums)
        {
            freq.add(Collections.frequency(nums,i));
        }
        if(freq.size()==1)
        {
            System.out.print("false");
        }
        else
        {
            System.out.print("true");
        }
    }
}
```

	Input	Expected	Got	
✓	4 1 2 3 1	true	true	✓
✓	4 1 2 3 4	false	false	✓
✓	10 1 1 1 3 3 4 3 2 4 2	true	true	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 25.00

You are given an integer array nums. The unique elements of an array are the elements that appear exactly once in the array.

Return the sum of all the unique elements of nums.

Example 1:

Input:

4

1 2 3 2

Output: 4

Explanation: The unique elements are [1,3], and the sum is 4.

Example 2:

Input:

5

1 1 1 1 1

Output: 0

Explanation: There are no unique elements, and the sum is 0.

Example 3:

Input:

5

1 2 3 4 5

Output: 15

Explanation: The unique elements are [1,2,3,4,5], and the sum is 15.

Constraints:

1 <= nums.length <= 100

1 <= nums[i] <= 100

For example:

Input	Result
4 1 2 3 2	4
5 1 1 1 1 1	0
5 1 2 3 4 5	15

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class SumOfUniqueElements
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int sum=0;
        ArrayList<Integer> nums = new ArrayList<Integer>();
        for(int i=0;i<n;i++)
        {
            nums.add(sc.nextInt());
        }
        for(int i=0;i<n;i++)
        {
            if((Collections.frequency(nums,nums.get(i)))==1)
            {
                sum+=nums.get(i);
            }
        }
        System.out.print(sum);
    }
}
```

	Input	Expected	Got	
✓	4 1 2 3 2	4	4	✓
✓	5 1 1 1 1 1	0	0	✓
✓	5 1 2 3 4 5	15	15	✓

Passed all tests! ✓

Question **4**

Correct

Marked out of 25.00

A pangram is a sentence where every letter of the English alphabet appears at least once.

Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise.

Example 1:

Input: sentence = "thequickbrownfoxjumpsoverthelazydog"

Output: true

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

Input: sentence = "rajalakshmi"

Output: false

Constraints:

1 <= sentence.length <= 1000

sentence consists of lowercase English letters.

For example:

Input	Result
thequickbrownfoxjumpsoverthelazydog	true
rajalakshmi	false

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class Pangram
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        Set<Character> hs=new HashSet<Character>();
        for(int i=0;i<s.length();i++)
        {
            hs.add(s.charAt(i));
        }
        if(hs.size()==26)
        {
            System.out.print("true");
        }
        else
        {
            System.out.print("false");
        }
    }
}
```

	Input	Expected	Got	
✓	thequickbrownfoxjumpsoverthelazydog	true	true	✓
✓	rajalakshmi	false	false	✓

Passed all tests! ✓