1. At a fun fair a street vendor is selling different colors of ballons. He sells N number of different colors of ballons (B[]). The task is to find the color (odd) of the ballon which is present odd number of times in the bunch of balloons.

Note.If there is more than one color which is odd in number then the first color in the array which is present odd number of times is displayed.The colour of the ballons can all either upper case or lower case in the array. If the inputs are even in the number, display the message “All are Even”.

* 🡪 **Approach**:
  + Use two loops to count the frequency of each balloon color.
  + Check which balloon color has an odd count.
  + Return the first balloon with an odd count.
* **Time Complexity**: O(N2)O(N^2)O(N2)
* **Space Complexity**: O(1)O(1)O(1)

**Code (Brute Force)**

import java.util.Scanner;

public class BalloonOddFinder {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of balloons: ");

int n = sc.nextInt();

sc.nextLine(); // Consume newline

char[] balloons = new char[n];

System.out.print("Enter balloon colors (single character each): ");

for (int i = 0; i < n; i++) {

balloons[i] = sc.next().charAt(0);

}

String result = findOddBalloonBruteForce(balloons);

System.out.println(result);

}

public static String findOddBalloonBruteForce(char[] balloons) {

int n = balloons.length;

boolean allEven = true;

for (int i = 0; i < n; i++) {

int count = 0;

for (int j = 0; j < n; j++) {

if (balloons[i] == balloons[j]) {

count++;

}

}

if (count % 2 != 0) {

return "First odd balloon color: " + balloons[i];

}

}

return "All are Even";

}

}

**Approach**

1. Use a **HashMap** to store the frequency of each balloon color.
2. Traverse the array once to count the occurrences of each balloon.
3. Traverse the array again to find the **first balloon** with an **odd count**.
4. If no balloon has an odd count, print "All are Even".

import java.util.HashMap;

import java.util.Scanner;

public class BalloonOddFinder {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of balloons: ");

int n = sc.nextInt();

sc.nextLine(); // Consume newline

char[] balloons = new char[n];

System.out.print("Enter balloon colors (single character each): ");

for (int i = 0; i < n; i++) {

balloons[i] = sc.next().charAt(0);

}

String result = findOddBalloon(balloons);

System.out.println(result);

}

public static String findOddBalloon(char[] balloons) {

HashMap<Character, Integer> frequencyMap = new HashMap<>();

// Count frequencies using HashMap

for (char balloon : balloons) {

frequencyMap.put(balloon, frequencyMap.getOrDefault(balloon, 0) + 1);

}

// Find the first balloon with an odd count

for (char balloon : balloons) {

if (frequencyMap.get(balloon) % 2 != 0) {

return "First odd balloon color: " + balloon;

}

}

return "All are Even";

}

}

1. Find the unique element in the array there will be only one unique element in the array x=[5,3,2,3,2] o/p=5.
2. public class UniqueElementInArray {  
     
    public static void main(String[] args) {  
    int[] arr = {5, 3, 2, 3, 2};  
    int uniqueElement = -1; // Initialize with -1 to indicate no unique element found  
     
    for (int i = 0; i < arr.length; i++) {  
    boolean isUnique = true;  
     
    for (int j = 0; j < arr.length; j++) {  
    if (i != j && arr[i] == arr[j]) {  
    isUnique = false;  
    break; // No need to check further if a duplicate is found  
    }  
    }  
     
    if (isUnique) {  
    uniqueElement = arr[i];  
    break; // Exit once the first unique element is found  
    }  
    }  
     
    if (uniqueElement == -1) {  
    System.*out*.println("No Unique Element Found");  
    } else {  
    System.*out*.println("Unique Element in an Array is: " + uniqueElement);  
    }  
    }  
   }

//using Hashmap  
public static void main(String[] args) {  
 int[] arr = {5, 3, 2, 3, 2,5,8};  
 int uniqueElement = -1; // Initialize with -1 to indicate no unique element found  
  
 // Step 1: Count frequencies using HashMap  
 HashMap<Integer, Integer> frequencyMap = new HashMap<>();  
 for (int num : arr) {  
 frequencyMap.put(num, frequencyMap.getOrDefault(num, 0) + 1);  
 }  
  
 // Step 2: Find the first unique element  
 for (int num : arr) {  
 if (frequencyMap.get(num) == 1) {  
 uniqueElement = num;  
 break;  
 }  
 }  
  
 // Step 3: Display the result  
 if (uniqueElement == -1) {  
 System.*out*.println("No Unique Element Found");  
 } else {  
 System.*out*.println("Unique Element in an Array is: " + uniqueElement);  
 }  
}

Greedy Algorithm…\*\*\*\*\*?.........