

TASK 1

AIM: To check whether the two halves of a given string contain an equal number of vowels.

ALGORITHM

1. Read the input string s.
2. Find the length of the string and divide it into two equal halves.
3. Initialize two counters:
 - countA for vowels in the first half
 - countB for vowels in the second half
4. Traverse the first half of the string:
 - If a character is a vowel, increment countA.
5. Simultaneously check the corresponding character in the second half:
 - If it is a vowel, increment countB.
6. Compare countA and countB.
7. If both counts are equal, return true; otherwise, return false.

PROGRAM

```
import java.util.Scanner;

class Solution {

    public boolean halvesAreAlike(String s) {
        int n = s.length();
        int mid = n / 2;

        int countA = 0;
        int countB = 0;

        for (int i = 0; i < mid; i++) {
            if (isVowel(s.charAt(i))) {
                countA++;
            }
            if (isVowel(s.charAt(i + mid))) {
                countB++;
            }
        }

        return countA == countB;
    }

    private boolean isVowel(char ch) {
        ch = Character.toLowerCase(ch);
        return ch == 'a' || ch == 'e' || ch == 'i' ||
            ch == 'o' || ch == 'u';
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a string:");
        String s = sc.nextLine();

        Solution obj = new Solution();
```

```
boolean result = obj.halvesAreAlike(s);

if (result) {
    System.out.println("Both halves have equal number of vowels.");
} else {
    System.out.println("Both halves do not have equal number of vowels.");
}

sc.close();
}
}
```

OUTPUT

Accepted Runtime: 0 ms

✓ Case 1

✓ Case 2

Input

s =
"book"

Output

true

Expected

true

RESULT :The program successfully determines whether both halves of the string contain the same number of vowels.

TASK 2

AIM: To check whether a given string is a Lapindrome.

ALGORITHM:

1. Read the input string s.
2. Find the length of the string n.
3. Create two integer arrays freq1 and freq2 of size 26 to store character frequencies.
4. Calculate the middle index mid = n / 2.
5. Store the frequency of characters from index 0 to mid - 1 in freq1.
6. If the string length is odd, skip the middle character.
7. Store the frequency of characters from the second half of the string in freq2.
8. Compare both frequency arrays:
 - If all frequencies match, the string is a Lapindrome.
 - Otherwise, it is not a Lapindrome.
9. Print "YES" if the string is a Lapindrome, else print "NO".

PROGRAM

```
import java.util.*;
import java.lang.*;
import java.io.*;
class Codechef {
    public static void main (String[] args) throws java.lang.Exception {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        while (T-- > 0) {
            String s = sc.next();
            int n = s.length();
            int[] freq1 = new int[26];
            int[] freq2 = new int[26];
            int mid = n / 2;

            for (int i = 0; i < mid; i++) {
                freq1[s.charAt(i) - 'a']++;
            }
            for (int i = (n % 2 == 0 ? mid : mid + 1); i < n; i++) {
                freq2[s.charAt(i) - 'a']++;
            }
            boolean isLapindrome = true;
            for (int i = 0; i < 26; i++) {
                if (freq1[i] != freq2[i]) {
                    isLapindrome = false;
                    break;
                }
            }
            if (isLapindrome)
                System.out.println("YES");
            else
                System.out.println("NO");
        }
        sc.close();
    }
}
```

OUTPUT:

Input

```
3
gaga
abcba
abca
```

Output

```
objectivec
YES
YES
NO
```

RESULT:The program successfully determines whether each given string is a Lapindrome.

