

# AtCoder Beginner Contest 109

Problema B: *Shiritori*

---

Prof. Edson Alves - UnB/FGA

# **AtCoder Beginner Contest 109 – Problem B: Shiritori**

---

# Problema

Takahashi is practicing *shiritori* alone again today.

Shiritori is a game as follows:

- In the first turn, a player announces any one word.
- In the subsequent turns, a player announces a word that satisfies the following conditions:
  - That word is not announced before.
  - The first character of that word is the same as the last character of the last word announced.

In this game, he is practicing to announce as many words as possible in ten seconds.

You are given the number of words Takahashi announced,  $N$ , and the  $i$ -th word he announced,  $W_i$ , for each  $i$ . Determine if the rules of shiritori was observed, that is, every word announced by him satisfied the conditions.

## Constraints

- $N$  is an integer satisfying  $2 \leq N \leq 100$ .
- $W_i$  is a string of length between 1 and 10 (inclusive) consisting of lowercase English letters.

## Input

Input is given from Standard Input in the following format:

$$N$$
$$W_1$$
$$W_2$$
$$\vdots$$
$$W_N$$

### Output

If every word announced by Takahashi satisfied the conditions, print 'Yes'; otherwise, print 'No'.

# Exemplo de entradas e saídas

## Exemplo de Entrada

4  
hoge  
english  
hoge  
enigma  
  
9  
basic  
c  
cpp  
php  
python  
nadesico  
ocaml  
lua  
assembly

## Exemplo de Saída

No

Yes

## Solução

- A solução do problema consiste em observar as duas regras básicas do jogo
- Para tal, seja  $s_1$  a primeira palavra dita e  $c$  o último caractere de  $s$
- Para manter a primeira regra, as palavras já ditas devem ser mantidas em um conjunto
- Logo  $s_1$  deve ser inserida neste conjunto
- Para as demais palavras  $s_i$ , com  $2 \leq i \leq N$ , deve-se verificar primeiramente a segunda regra, isto é, checar se o primeiro caractere de  $s_i$  é igual a  $c$
- Caso não seja, a resposta do problema é 'No'
- Em caso afirmativo,  $s_i$  deve ser inserida no conjunto: caso já esteja lá, a resposta do problema também é 'No'
- Se não estiver no conjunto, o valor de  $c$  deve ser atualizado para o último caractere de  $s_i$  e continuar a rotina
- Esta solução tem complexidade  $O(N \log N)$ , devido as inserções no conjunto

## Solução AC com complexidade $O(N \log N)$

```
1 #include <bits/stdc++.h>
2
3 bool solve(int N, const std::vector<std::string>& words)
4 {
5     auto c = words[0].back();
6
7     std::set<std::string> S;
8     S.emplace(words[0]);
9
10    for (int i = 1; i < N; ++i)
11    {
12        if (S.count(words[i]) or words[i].front() != c)
13            return false;
14
15        c = words[i].back();
16        S.emplace(words[i]);
17    }
18
19    return true;
20 }
```



## Solução AC com complexidade $O(N \log N)$

```
22 int main()
23 {
24     std::ios::sync_with_stdio(false);
25
26     int N;
27     std::cin >> N;
28
29     std::vector<std::string> words(N);
30
31     for (int i = 0; i < N; ++i)
32         std::cin >> words[i];
33
34     auto ans = solve(N, words);
35
36     std::cout << (ans ? "Yes" : "No") << '\n';
37
38     return 0;
39 }
```