

Problem: Watto and mechanism.

Approach:

For every string:

- (i) Iterate through every stored string 't' with the same length.
- (ii) Compare 's' and 't' character by character
- (iii) Count mismatches
- (iv) If exactly one mismatch  $\rightarrow$  return Yes, else NO

Brute force:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    int n, m;
```

```
    cin >> n >> m;
```

IMPORTANT NOTES

```
vector<string> dict(n);
```

```
for(int i=0; i<n; i++) cin >> dict[i];
```





DATE: .....

```
while(m--)&
```

```
String s;
```

```
cin >> s;
```

```
bool isOk = false;
```

```
for(int i=0; i<n && !isOk; i++)&
```

```
if (cin.get().size() != s.size()) continue;
```

```
int diff = 0;
```

```
for(int j=0; j<size(); j++)&
```

```
if (s[j] != cin.get()[j]) diff++;
```

```
if (diff > 1) break;
```

```
&
```

```
if (diff == 1) isOk = true;
```

```
&
```

```
cout << (isOk ? "yes" : "no");
```

```
&
```

```
return 0;
```

```
&
```

### IMPORTANT NOTES

Time Complexity  $\rightarrow O(n \times m \times L)$





DATE: .....

Optimal:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
using ull = unsigned long long;
```

```
int main() {
```

```
    const ull BASE = 911382323ull;
```

```
    int n, m;
```

```
    cin >> n >> m;
```

```
    vector<unordered_set<ull>> s(600001);
```

```
    vector<ull> pw(600001);
```

```
    pw[0] = 1;
```

```
    for (int i = 1; i < pw.size(); i++) {
```

```
        pw[i] = pw[i-1] * BASE;
```

```
        auto val = [&](char c) -> ull {
```

```
            return (ull)(c - 'a' + 1);
```

```
        };
```

```
        auto getHash = [&](const string &s) -> ull {
```

IMPORTANT NOTES

```
            ull h = 0;
```

```
            for (int i = 0; i < s.size(); i++) {
```

```
                h += val(s[i]) * pw[i];
```

```
            };
```

```
            return h;
```





DATE: .....

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

```
    string s;
```

```
    cin >> s;
```

```
    St[s.size()].insert(getHash(s));
```

```
}  
while (m-- > 0)
```

```
    string s;
```

```
    cin >> s;
```

```
    int L = (int) s.size();
```

```
    ull h = getHash(s);
```

```
    bool isOk = false;
```

```
    for (int i = 0; i < L && !isOk; i++) {
```

```
        ull cur = val(s[i]) * pw[i];
```

```
        for (char c = 'a'; c <= 'z'; c++) {
```

```
            if (c == s[i]) continue;
```

```
            ull nh = h - cur + val(c) * pw[i];
```

```
            if (St[L].find(nh) != St[L].end()) {
```

IMPORTANT NOTES

```
                isOk = true;
```

```
                break;
```

```
            }  
        }  
    }
```

```
    cout << (isOk ? "yes" : "no");
```

Time Complexity:  $O((n+m)*L)$



ELEGANT

INDIA'S no.1 QST BAR