Name:-) Asmit Prabhakar
Uid :-) 23BCS10063
Class :-) KRG 2B.

Subject
-) Competitive coding 2.

Problem :- Watto and mechanism.

Statement :- Given strings s, determine if the memory of the mechanism contains string t that consist of same number of characters and s and differs froms in exactly one position.

Algorithm/approach :-
~~Brute force~~ :-

1. Itreate through every sorted string 't' with same length.
2. Compare 's' and 't' character by character.
3. count mismatches. happens
4. If the mismatch ==1, then return true otherwise return false.

Brute force :-

```
#include<bits/stdc++.h>
using namespace std;
 int main(){
   int m, n;
    cin>>n>>m;
   vector<string>dist(n);
   for(i=0; i<n; i++){
        cin>>dist[i];
      }
      while(m--){
        String s;
        cin>>s;
```

```
bool isok = false;
for(int i = 0; i< n&&!isok; i++){
    if((int) dict.size()! = s.size()){
            continue;
          int diff = 0;
        for(int j = 0; j< size(); j++){
            if(s[j]! = dict[i][j]){
                diff++;
                ~~&~~ if(diff >1){
                        break;

            }
            if(diff = = 1) {
                isok = true;
              }
            cout<<(isok? "yes": "No");
            }
            return 0;
        }
    T.C = (m x n x L).
```

optimal :-

```
    #include<bits/stdc++.h>
        using namespace std;
        using   ull = unsigned long long;
```

_/_/_

```
int main(){
    const ullBASE = 911382323ull;
      int m, n;
        cin>m;
        cin>n;
      vector<unordered_set<ull> st(600001);
       vector<ull> bw(600001);
         pw[0] = 1;
       for(int i = ~~0~~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{
            ull h = 0;
          for(int i = 0; i<s.size(); i++){
              h+ = val(s[i]) * pw[i];
          }
            return h;
        }
      for(int i = 0; i<n; i++){
            string s;
              cin>>s;
          st[s.size()].insert(getHash(s));
      }
            while(m--){
                 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 <= 'c'; c++){
         if(c == s[i])
             continue;
         ull nh = h - cur + val(c) * pw[i];
         if(st[L].find(nh) != st[L].end()){
               isok = true;
                 break;
         }
       }
    }
    cout << (isok ? "YES" : "NO");
  }
  T.C = O((n+m)*L).
```

165C