

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

```
using namespace std;
```

```
vector < pair <int, int> > p_pow;
```

```
const int mod1 = 1e9 + 7, mod2 = 1'000'000'009;
```

```
int p;
```

```
void make_pows(string &s){
```

```
    int n = s.size() + 1;
```

```
    p_pow.resize(n);
```

```
    p_pow[0] = {1, 1};
```

```
    for(int i = 1; i < n; ++i){
```

```
        p_pow[i].first = (p_pow[i - 1].first * 1ll * p) % mod1;
```

```
        p_pow[i].second = (p_pow[i - 1].second * 1ll * p) % mod2;
```

```
    }
```

```
}
```

```
void make_hashes(vector < pair <int, int> > &h, string &s){
```

```
    int n = s.size() + 1;
```

```
    h.resize(n);
```

```
    h[0] = {0, 0};
```

```
    for(int i = 1; i < n; ++i){
```

```
        h[i].first = (h[i - 1].first * 1ll * p + s[i - 1]) % mod1;
```

```
        h[i].second = (h[i - 1].second * 1ll * p + s[i - 1]) % mod2;
```

```
    }
```

```
}
```

```
pair <int, int> get_hash(vector < pair <int, int> > &h, int l, int r){
```

```
    pair <int, int> ans;
```

```
    ans.first = (h[r + 1].first - h[l].first * 1ll * p_pow[r - l + 1].first) % mod1;
```

```
    if (ans.first < 0){
```

```

        ans.first += mod1;
    }
    ans.second = (h[r + 1].second - h[l].second * 1ll * p_pow[r - l + 1].second) % mod2;
    if (ans.second < 0){
        ans.second += mod2;
    }
    return ans;
}

```

```

bool equal(vector < pair <int, int> > &h, int l1, int r1, int l2, int r2){
    if (r2 - l2 != r1 - l1){
        return 0;
    }
    return get_hash(h, l1, r1) == get_hash(h, l2, r2);
}

```

```

int strcmp(vector < pair <int, int> > &h, string &s, int l1, int r1, int l2, int r2){
    // 0 - подстроки равны
    // -1 - первая меньше второй
    // 1 - первая больше второй
    int len1 = r1 - l1 + 1, len2 = r2 - l2 + 1, len = min(len1, len2);
    int l = 0, r = len + 1;
    while (r - l > 1){
        int m = (l + r) / 2;
        if (get_hash(h, l1, l1 + m - 1) == get_hash(h, l2, l2 + m - 1)){
            l = m;
        }
        else {
            r = m;
        }
    }
    if (l == len){ // префиксы полностью совпали

```

```

        if (len1 < len2){
            return -1;
        }
        if (len1 == len2){
            return 0;
        }
        if (len1 > len2){
            return 1;
        }
    }
    if (s[l1 + l] < s[l2 + l]){
        return -1;
    }
    return 1;
}

```

```

bool cmp(vector < pair <int, int> > &h, string &s, int l1, int r1, int l2, int r2){
    // 1 - первая подстрока меньше второй
    // 0 - в противном случае
    int len1 = r1 - l1 + 1, len2 = r2 - l2 + 1, len = min(len1, len2);
    int l = 0, r = len + 1;
    while (r - l > 1){
        int m = (l + r) / 2;
        if (get_hash(h, l1, l1 + m - 1) == get_hash(h, l2, l2 + m - 1)){
            l = m;
        }
        else {
            r = m;
        }
    }
    if (l == len){ // префиксы полностью совпали
        return len1 < len2;
    }
}

```

```
    }  
    return s[l1 + l] < s[l2 + l];  
}
```

```
int main(){  
    ios_base::sync_with_stdio(0);  
    cin.tie(0);  
    string s, t;  
    cin >> s >> t;  
    int x = rand(), n = s.size(), m = t.size();  
    p = max(257, x + (x % 2 == 0));  
    vector < pair <int, int> > h_s, h_t;  
    make_pows(s);  
    make_hashes(h_s, s);  
    make_hashes(h_t, t);  
}
```