**Longest Distinct characters in string**

[string](http://www.practice.geeksforgeeks.org/tag-page.php?tag=string&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)

Given a string, find length of the smallest substring with maximum number of distinct characters.  For example, for input "abca", the output is 3 as "abc" is the smallest substring with maximum number of distinct characters.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is String str.  
  
**Output:**

Print length of smallest substring with maximum number of distinct characters.  
  
**Constraints:**

1 ≤ T ≤ 100  
1 ≤ size of str ≤ 10000  
  
**Example:**

Input  
2  
abababcdefababcdab  
geeksforgeeks

Output:  
6  
7

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=412>

#include <iostream>

#include <stdio.h>

#include <set>

using namespace std;

int main() {

    // TODO code application logic here

    int t;

    scanf("%d", &t);

    while(t-- > 0) {

        std::string s;

        cin >> s;

        int max\_len = 0;

*/\* recorro cada caracter de la cadena \*/*

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

            std::set<char> hs;

*/\* recorro desde i hasta el final de la cadena \*/*

            for (int j = i; j < s.length(); j++) {

*/\*si el set no contiene s[j], lo agrego al set\*/*

                if (!hs.count(s[j])) {

                   // x is in the set, count is 1

                   hs.insert(s[j]);

                } else { */\* de lo contrario si esta en set*

*es porque tengo que parar, verificar si es la subcadena mas larga,*

*y empezar desde i \*/*

                   // count zero, i.e. x not in the set

                   break;

                }

            }

            if (hs.size() > max\_len) {

                max\_len = hs.size();

            }

        }

        printf("%d**\n**", max\_len);

    }

}