**Solitary String**

Max. Marks: 100

In order to celebrate Women's day, Stella started playing with string SS. She needs to know the maximum number of characters between any 22 same characters in the string.

As she is busy in playing, help her for the same.

If there are no 22 same characters in the string, print −1−1.

Note: String is composed of lowercase letters of the Latin alphabet.

***Input:***  
First line contains one integer TT, denoting the number of test cases.   
Each of the next TT line contains one string SS.

***Output:***  
For each test case, output the maximum number of characters between any 22 same characters in the string. If there are no 22 same characters in the string, print −1−1.   
Print answer for each test case in a new line.

***Constraints:***  
1≤T≤101≤T≤10   
1≤|S|≤1051≤|S|≤105, where |S||S| determines the length of the string.  
String is composed of lowercase alphabets ranging from aa to zz.

**SAMPLE INPUT**

2

aba

babcddc

**SAMPLE OUTPUT**

1

2

**Explanation**

Here, for string = abaaba

There is only one character between 2 occurrences of aa.

2) For string = babcddcbabcddc

There is one character between 2 occurrences of bb, and 2 characters between 2 occurrences of cc. So the answer is 22.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded if any testcase passes.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/challenge/competitive/international-women-hackathon-2017/algorithm/solitary-string/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

// string s = "babcddc";

//string s = "aabcde";

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

string s = Console.ReadLine();

Dictionary<char, int[]> diccio = new Dictionary<char, int[]>();

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

{

if (diccio.ContainsKey(s[i]))

{

diccio[s[i]] = new int[] { diccio[s[i]][0], i };

}

else

{

diccio[s[i]] = new int[] { i, i };

}

}

int max\_dif = 0;

char max\_char = ' ';

foreach (KeyValuePair<char, int[]> kvp in diccio)

{

//Console.WriteLine(kvp.Key + " -> " + (kvp.Value[1] - kvp.Value[0] + 1));

int dif = kvp.Value[1] - kvp.Value[0] + 1;

if (dif > max\_dif)

{

max\_dif = dif;

max\_char = kvp.Key;

}

}

if (max\_dif == 0)

{

Console.WriteLine(-1);

}

else

{

Console.WriteLine(max\_dif - 2);

}

}

Console.ReadLine();

}

}

}