Problem: Anagram

Sid is obsessed with reading short stories. Being a CS student, he is doing some interesting frequency analysis with the books. He chooses strings  and  in such a way that .

Your task is to help him find the minimum number of characters of the first string he needs to change to enable him to make it an [anagram](https://en.wikipedia.org/wiki/Anagram) of the second string.

*Note:* A word *x* is an anagram of another word *y* if we can produce *y* by rearranging the letters of *x*.

**Input Format**

The first line will contain an integer, , representing the number of test cases. Each test case will contain a string having length , which will be concatenation of both the strings described above in the problem.The given string will contain only characters from  to .

**Constraints**



**Output Format**

An integer corresponding to each test case is printed in a different line, i.e. the number of changes required for each test case. Print  if it is not possible.

**Sample Input**

6

aaabbb

ab

abc

mnop

xyyx

xaxbbbxx

**Sample Output**

3

1

-1

2

0

1

**Explanation**

*Test Case #01:* We have to replace all three characters from the first string to make both of strings anagram. Here,  = "aaa" and  = "bbb". So the solution is to replace all character 'a' in string *a* with character 'b'.   
  
*Test Case #02:* You have to replace 'a' with 'b', which will generate "bb".   
  
*Test Case #03:* It is not possible for two strings of unequal length to be anagram for each other.   
  
*Test Case #04:* We have to replace both the characters of first string ("mn") to make it anagram of other one.   
  
*Test Case #05:*  and  are already anagram to each other.   
  
*Test Case #06:* Here *S1 = "xaxb"* and *S2 = "bbxx"*. He had to replace 'a' from *S1* with 'b' so that *S1 = "xbxb"* and we can rearrange its letter to "bbxx" in order to get *S2*.

Solution

int main() {

string str, strs="";

int cases, length;

cin>>cases;

for(int i=0; i<cases; i++)

{

int count = 0;

int score[26] = {0};

cin>>str;

length = str.length();

if(length%2 == 0) //even length

{

//making two different strings

length = length / 2;

strs = str.substr (0, length);

str = str.substr (length, length\*2);

//traversing the second string

for(int j=0; j<length; j++)

{

int temp = (int)strs[j] -97 ;

score[temp]+=1;

}

//counting the operations required

for(int j=0; j<length; j++)

{

int temp = (int)str[j] - 97;

if(score[temp]>0)

{ score[temp]-=1; }

else

{ count+=1; }

}

//printing the result

cout<<count<<endl;

}

else

{ cout<<"-1"<<endl; }

}

return 0;

}

Elegant

int main() {

string str, strs="";

int cases, length;

cin>>cases;

for(int i=0; i<cases; i++)

{

cin>>str;

int count = 0;

length = str.length();

if(length%2 !=0 ) cout<<"-1"<<endl;

else

{

for(int j=0; j<length/2; j++)

{

for(int k=length/2; k<length; k++)

{

if(str[j]==str[k])

{

count+=1;

str[k]='\*';

break;

}

}

}

cout<<(length/2) -count<<endl;

}

}

return 0;

}

`’Anshul AgGarwal