Given a string S, your task is to find its substring that occurs most often in S. Among all such substrings you must find the one which has the maximum length. As a result return only one integer - the length of the answer.

**Example:**

For S = "abab" the output should be 2.

* **[input] string S**
  + 2 ≤ s.length ≤ 200
* **[output] integer**

<https://codefights.com/challenge/QaiYqG6Q3wtsRrJi2>

#include<iostream>

#include<conio.h>

#include <vector>

#include <map>

#include <conio.h>

#include <stdio.h>

using namespace std;

int strings(std::string S) {

    std::map<std::string, int> m;

    int maxim = 0;

*/\* busco todos los substrings posibles desde i hasta j,*

*los guardo en un hashmap para saber la cantidad de veces que esta cada uno \*/*

    for(int i = 0; i < S.size(); i++)

    {

        std::string subs = "";

        for(int j =i; j < S.size(); j++){

            subs = S.substr(i, j-i +1);

            m[subs]++;

*/\* obtengo el maximo de las veces que esta un substring\*/*

            maxim = std::max(maxim, m[subs]);

        }

    }

    int maxLen = 0;

*/\* como substrings distintos pueden estar la misma cantidad de veces*

*del maximo obtengo la cadena de longitud mas larga\*/*

    for(std::map<std::string, int>::iterator it = m.begin(); it != m.end(); it++) {

        if(it->second == maxim){

            if(it->first.length() > maxLen) {

                maxLen = it->first.length();

            }

        }

    }

    return maxLen;

}

int main(){

     std::string s ="abab";

   int res = strings(s);

   printf("%d ", res );

    getch();

    return 0;

}