using System;

using System.Globalization;

using System.Linq;

using static System.Console;

namespace Homework29.\_08.\_23

{

internal class Program

{

//1

/\*static void Main(string[] args)

{

NumberFormatInfo numberformatinfo = new NumberFormatInfo()

{

NumberDecimalSeparator = "."

};

float x, y, z;

WriteLine("Enter x and y");

x = Convert.ToSingle(Console.ReadLine(), numberformatinfo);

y = Convert.ToSingle(Console.ReadLine(), numberformatinfo);

z = (float)((Math.Sin(x) + (2 + y) / (1 + (float)Math.Pow(x, 2))) / ((float)Math.Sqrt(Math.Abs(y)) + 2));

WriteLine("x = {0:f4}, y = {1}, z = {2:f6}", x, y, z);

}\*/

//2

/\*

static void Main(string[] args)

{

\*//\*NumberFormatInfo numberformatinfo = new NumberFormatInfo()

{

NumberDecimalSeparator = "."

};\*//\*

int EnterX, EnterY;

WriteLine("Enter X Y");

int[] mass = new int[6];

for (int i =0,j = 0; i < 3; i++)

{

EnterX = Int32.Parse(Console.ReadLine());

EnterY = Int32.Parse(Console.ReadLine());

mass[j] = EnterX;

j++;

mass[j]=EnterY;

j++;

}

int side;

int side2;

int side3;

//a

int x1 = mass[0];

int y1 = mass[1]; //можно было просто проинициализировать переменные через Console.ReadLine() я не должен был создавать столько перемен но собака директива #define не работает а целью создание перемен чтоб не путаться с mass[0,1,2,3,4,5] можно было сделать двух мерный массив но мне лень

//b

int x2 = mass[2];

int y2 = mass[3];

//c

int x3 = mass[4];

int y3 = mass[5];

side = (int)Math.Abs(Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2) ) );

side2 = (int)Math.Abs(Math.Sqrt(Math.Pow(x3 - x1, 2) + Math.Pow(y3 - y1, 2)));

side3 = (int)Math.Abs(Math.Sqrt(Math.Pow(x3 - x2, 2) + Math.Pow(y3 - y2, 2)));

int P = side + side2 + side3;

int p = P / 2;

int S = (int)(Math.Sqrt(p \* (p - side) \* (p - side2) \* (p - side3)));

int Pif = (int)(Math.Pow(side, 2)+ Math.Pow(side2,2)); //и по сути должно найтись Pow(side3,2) но такая фигня что данный треугольник по кординатам не всегда может быть прямоугольным поэтому тут будет баг чтобы исправить его надо сделать проверку на прямоугольный вообще треугольный но мне лень этим заниматься, тай вообще у нас и так изначально три стороны известны поэтому можно просто написать pow(side3,2) и вот результат нахождение гипотенузы

WriteLine("sides: {0} {1} {2}", side, side2, side3);

WriteLine("P: "+P+" p: "+p);

WriteLine("S: "+S);

WriteLine("Pif: " + Pif);

}\*/

static void Main(string[] args)

{

string s=ReadLine();

string set=" ";

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

{

if (i == 0)

{

set = s[i].ToString();

}

else

{

bool flag = true;

for(int j =0;j<set.Length;j++)

{

if (set[j] != s[i]) //aaabbbccc a

{

flag = false;

}

else

{

flag=true; break;

}

}

if (!flag)

set += s[i].ToString();

}

}

int[]mass = new int[set.Length];

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

{

mass[i] = s.Count(c => c == set[i]);

}

foreach(int i in mass)

Console.WriteLine(i);

int max = mass[0];

int maxindex = 0;

for(int i=1;i< mass.Length; i++)

{

if (mass[i] > max)

{

max = mass[i];

maxindex = i;

}

}

Console.WriteLine("найбільше всього символів: " + set[maxindex] + " всього їх :" + max);

}

}

}