Caja Negra...

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Descipción**

Como ya es sabido Karel1Karel1 falleció en un desafortunado accidente hace poco, pero su hermano Karel2Karel2siente la necesidad de saber a ciencia cierta qué fue lo que pasó y para ello planea utilizar la caja negra que llevaba consigo en el camión Karel1Karel1.

**Problema**

Ayuda a Karel2Karel2 a saber cuál es la contraseña para abrir la caja negra y tener acceso a las grabaciones. La caja tiene una contraseña que consiste en la letra que más se repite de una cadena de caracteres(palabra) que está inscrita en ella, tu tarea consiste en decirnos cuál es la letra que más se repite(verver ejemploejemplo).

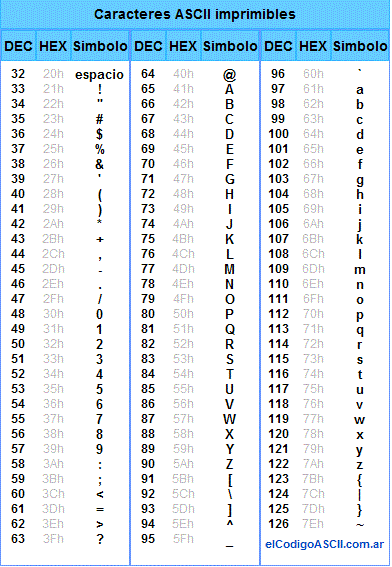
**Entrada**

Una línea con N1N1, la longitud de la palabra de la caja, después una línea con una palabra que es la que contiene la contraseña.

**Salida**

Una línea con una letra la cual indica la letra que más se repite en la palabra.

**Códigos ASCII**



**Ejemplo**

|  |  |  |
| --- | --- | --- |
| 7  kareles | e | En la palabra se repite más la letra e. |

**Límites**

* La longitud de la palabra no será mayor a 30.
* La palabra solo tendrá letras minúsculas.

*Fuente: Cope*

Problema subido por: [DelegadoSonora](https://omegaup.com/profile/DelegadoSonora/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Caja-Negra#problems/new-run) | | | | | | | | |
| b4636bdd | Respuesta correcta | 100.00% | 0 | cpp | 3.11 | 0.01 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Caja-Negra#problems>

#include <iostream>

#include <stdio.h>

#include <map>

using namespace std;

int main()

{

int N;

scanf("%d", &N);

std::string s;

cin >> s;

std::map<char, int> m;

for(int i = 0; i < N; i++) {

m[s[i]]++;

}

char max\_char;

int max = 0;

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

if(it->second > max) {

max = it->second;

max\_char = it->first;

}

}

//printf("%d\n",max);

cout << max\_char << endl;

system("pause");

return 0;

}