

---

**Búsquedas web****X58286\_es**

---

La información de una búsqueda web la almacenamos en la siguiente estructura:

```
struct Search {
    string word;
    int stime;           // tiempo de respuesta
};
```

Dado un conjunto de búsquedas web S y un conjunto de palabras P decir, por cada palabra p en P, cuál ha sido el tiempo de la búsqueda web más lenta de esa palabra p en S.

Utiliza la estructura que te damos a continuación, introduciendo código SÓLO en los lugares indicados. En caso contrario el problema se considerará INVÁLIDO.

```
#include <iostream>
#include <vector>
using namespace std;

struct Search {
    string word;
    int stime;
};

// Pre: n >= 0
// Post: lee de la entrada una secuencia S de n búsquedas y retorna
//       esa información almacenada en un vector de Search
vector<Search> read_data(int n) {
    // tu código aquí
}

// Pre: -
// Post: retorna el tiempo de respuesta más lento de la palabra p
//       en v (si existe), o 0 en caso contrario
int tiempo_maximo(const vector<Search>& v, string p) {
    // tu código aquí
}

int main() {
    // tu código aquí
}
```

**Puntos examen: 2.50 Parte automática: 0.00%**

**Entrada**

La entrada consta de dos secuencias:

- La primera, la secuencia S de búsquedas es una secuencia de pares string p, int t que indican palabra buscada (en minúsculas y sin espacios en blanco) y su tiempo de respuesta, respectivamente. Antes de la secuencia S nos dirán el número de pares de la secuencia. **Implementa y usa la función read\_data para leer esta secuencia de la entrada.**
- A continuación, la secuencia P de palabras (en minúsculas y sin espacios en blanco).

## Salida

Para cada palabra p de la secuencia P, decir cuál ha sido su tiempo máximo de respuesta en las búsquedas de la secuencia S. Si p no aparece en S, entonces el tiempo máximo de respuesta será 0. Sigue el formato de los ejemplos.

### Ejemplo de entrada 1

```
7
pro1 3
algoritmo 5
pro1 1
procedimiento 3
procedimiento 10
recursivo 2
pro1 2

pro1
procedimiento
algoritmo
recursivo
nada
```

### Ejemplo de salida 1

```
pro1: 3
procedimiento: 10
algoritmo: 5
recursivo: 2
nada: 0
```

### Ejemplo de entrada 2

```
0

no
hay
búsquedas
web
```

### Ejemplo de salida 2

```
no: 0
hay: 0
búsquedas: 0
web: 0
```

### Ejemplo de entrada 3

```
3
sin 4
secuencia 6
palabras 8
```

### Ejemplo de salida 3

## Información del problema

Autoría: Pro1

Generación: 2026-01-25T17:06:44.430Z

© Jutge.org, 2006–2026.

<https://jutge.org>