



## Objetivos de la Guía

- Aprender sobre el uso de pares
- Aprender sobre el uso de Map

## > A - Paula y los registros

Paula quiere organizar un concurso, ella cree que es el momento perfecto para hacer uso de los Map.

### \* Parte 1: Por rut

Ella primero organizar el concurso de tal forma que dado un rut se puede determinar que número eligió para el concurso.

#### \* Input:

El input consiste en un entero  $n$ , que corresponde a la cantidad de personas en el concurso. Luego, le siguen  $n$  líneas que contienen un string  $r_i$  y un entero  $p_i$ ,  $r_i$  es el rut de la persona en la línea  $i$  y  $p_i$  es el número que eligió la persona  $i$ .

Luego, le sigue una línea que corresponde a un string  $r$ , este corresponde al string que uno quiere saber que número eligió.

```
4
11111111-1 10
1231234-2 9
12023832-1 5
12312124-k 3
12312124-k
```

#### \* Output:

Mostrar por consola el número de la persona consultada.

```
3
```

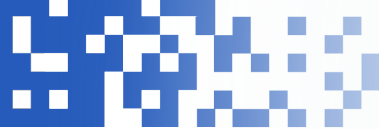
### Tip 1

Recuerde que para declarar un conjunto debe hacer `map< T1, T2 >`, dónde T1 se debe reemplazar por string y T2 se debe reemplazar por int.

### Tip 2

Recuerde que para insertar elementos en el Map debe crear un pair. Por ejemplo:

```
map< string, int > v;
v.insert(pair< string, int >("hola", 2));
// o tambien se puede hacer de la siguiente forma
v.insert({"hola", 2})
```



## \* Parte 2: Contar e imprimir

Paula sabiendo el poder de utilizar un Map, quiere saber cuantas veces un mismo rut participo en el concurso.

### \* Input:

Su programa recibirá un entero  $n$ , seguido de  $n$  líneas, dónde cada línea contiene un rut.

```
10
11111111-1
1231234-2
12023832-1
12023832-1
12023832-1
12312124-k
1231234-2
1231234-2
12312124-k
11111111-1
```

### \* Output:

Se debe imprimir los ruts de manera ordenada y seguido de la cantidad de veces que participa en el concurso.

```
11111111-1 2
12023832-1 3
12312124-k 2
1231234-2 3
```

### Tip 1

Recuerde que si usa `find`, retorna un iterador a un `pair` que tiene la llave y el valor. Y puedes acceder usando `->second` y ahí puedes modificar el valor directamente.

### Tip 2

Para recorrer los conjuntos debe hacer uso de los iteradores. Sigue este ejemplo (reemplace `T` y `T2` por el tipo de variable correspondiente):

```
// se asume que el nombre de la variable tipo map es x
for(map< T1, T2 >::iterator itr = x.begin(); itr != x.end(); itr++) {
    cout << itr->first << " " << itr->second << endl;
}
```