



## Objetivos de la Guía

- Aprender respecto a la búsqueda binaria
- Aprender sobre el uso de la búsqueda binaria implementada en C++

## > A - Gabriel no encuentra sus llaves

Gabriel tiene las llaves de su casa numeradas. Él las guarda en un plato. A veces el le cuesta mucho encontrar la llave de alguna de sus cerraduras, es por esto que quiere hacer un programa que reciba las llaves y diga en que posición está esa llave.

### \* Parte 1: Números son números

En esta primera parte, Gabriel le pone a cada llave un número. Te pide ayuda que si no su cabeza explotará.

#### \* Input:

La primera línea del input consiste en un entero  $n$  y  $k$ , que corresponde a la cantidad de llaves que tiene Gabriel y la llave a buscar. La siguiente línea contiene  $n$  enteros que corresponde a los números de las llaves.

```
5 2
0 2 3 5 8
```

#### \* Output:

Mostrar por pantalla la posición en que esta la llave.

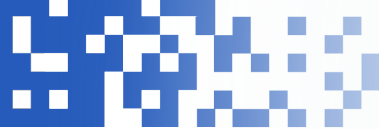
```
1
```

### Tip 1

Recuerde el uso de la función `lower_bound(x)`, que entrega un iterador al primer número que no sea menor que  $x$

```
// se asume que el vector v esta ordenado de menor a mayor
lower_bound(v.begin(), v.end(), x) // esto te entregará un iterador del primer elemento que no sea
↪ menor que x

lower_bound(v.begin(), v.end(), x) - v.begin() // esto te entregará la posición del elemento que
↪ encuentra
```



## \* Parte 2: Llaves con letras

Gabriel en vez de ponerle de números, ahora le puso letras. Le pide hacer lo mismo que el ejercicio anterior, pero con letras.

\* Input:

```
5 H
A C D G H
```

\* Output:

```
4
```

## \* Parte 3: Ayuda no se si tengo una llave

Ahora Gabriel, no le interesa saber en que posición esta la llave, sino que le interesa saber si la llave existe. Para colmo, un amigo de él le desordeno todas las llaves.

Para esta parte, las llaves siguen siendo letras.

\* Input:

```
5 H
H B C D A
```

\* Output:

Debe mostrar por pantalla "Existe" si la llave existe y "No Existe" si la llave no existe.

```
Existe
```

### Tip 1

Recuerde que para aplicar búsqueda binaria, los elementos deben estar ordenados. Utilice la función `sort` para ordenar.