

Introduction to Programming (in C++)

*Lectura de una secuencia de entrada
y
tratamiento de formato de salida*

Emma Rollón
Department of Computer Science

Leer una secuencia de entrada

- **Input:** La entrada es una secuencia de elementos.
- **Output:** Para cada uno de los elementos tenemos que realizar una tarea determinada.
- **Ejemplo:**
 - Input: La entrada es una secuencia de enteros.
 - Output: Para cada entero, queremos escribir su valor más uno. Cada número se escribirá en una línea acabada con final de línea.

Leer una secuencia de entrada

- El **formato** en el que nos dan esa secuencia de entrada determina el **esquema de lectura** a utilizar:
 1. Antes de darnos la secuencia, nos dicen el número de elementos que tendrá.
 2. El final de la secuencia viene marcada por un valor especial (que se denomina centinela)
 3. Directamente nos dan la secuencia.

Leer una secuencia de entrada

1. Antes de darnos la secuencia, nos dicen el número de elementos que tendrá:

E: 3 7 5 8

S: 8 ← 6 ← 9 ←

```
int n;
cin >> n; // Leemos el número de elementos de la secuencia
// Inv: Se han tratado i elementos de la secuencia
for (int i = 0; i < n; ++i) {
    // leer un elemento de la secuencia
    int x;
    cin >> x;
    // tratar el elemento que acabamos de leer
    cout << x + 1 << endl;
}
// Se han tratado todos los elementos de la secuencia
```

Leer una secuencia de entrada

1. Antes de darnos la secuencia, nos dicen el número de elementos que tendrá:

E: 3 7 5 8

S: 8 ← 6 ← 9 ←

- **Cuestiones:**
 - Qué entrada representa la secuencia vacía?
 - La entrada puede ser vacía?

Leer una secuencia de entrada

2. El final de la secuencia viene marcado por un **valor especial (centinela)**.

E: 7 5 8 -1

S: 8 ← 6 ← 9 ←

- **Observaciones:**

- El centinela es del mismo tipo que los elementos de la secuencia.
- El centinela no puede ser un elemento de la secuencia (si lo fuera, no sabría distinguir si es un elemento válido o el centinela).

Leer una secuencia de entrada

2. El final de la secuencia viene marcado por un **valor especial (centinela)**.

E: 7 5 8 -1

S: 8 ← 6 ← 9 ←

```
int x;
cin >> x; // Leemos el primer elemento de la entrada
// Inv: x es un elemento de la secuencia o es el centinela
while (x != -1) {
    // x es un elemento de la secuencia
    // tratar el elemento que acabamos de leer
    cout << x + 1 << endl;
    // leer el siguiente elemento de la entrada
    cin >> x;
}
// Se han tratado todos los elementos de la secuencia
// x es el centinela
```

Leer una secuencia de entrada

2. El final de la secuencia viene marcado por un **valor especial (centinela)**.

E: 7 5 8 -1

S: 8 ← 6 ← 9 ←

- **Cuestiones:**
 - Qué entrada representa la secuencia vacía?
 - La entrada puede ser vacía?

Leer una secuencia de entrada

3. La entrada es directamente la secuencia de elementos a tratar

E: 7 5 8

S: 8 ← 6 ← 9 ←

```
int x;
// Inv: x es un elemento de la secuencia o no hay más
//      elementos en la entrada
while (cin >> x) {
    // tratar el elemento que acabamos de leer
    cout << x + 1 << endl;
}
// Se han tratado todos los elementos de la secuencia
```

Leer una secuencia de entrada

3. La entrada es directamente la secuencia de elementos a tratar

E: 7 5 8

S: 8 ← 6 ← 9 ←

- **Cuestiones:**

- Qué entrada representa la secuencia vacía?
- La entrada puede ser vacía?
- Cómo se indica que la secuencia de entrada ha acabado?

Tratamiento formato de salida

- Y si cambiamos el formato de la salida?

E: 7 5 8 **S: 8, 6, 9** ←

- Cuál es la salida de este código?

```
int x;
// Inv: x es un elemento de la secuencia o no hay más
//      elementos en la entrada
while (cin >> x) {
    // tratar el elemento que acabamos de leer
    cout << x + 1 << \,\' ;
}
cout << endl;
// Se han tratado todos los elementos de la secuencia
```

Tratamiento formato de salida

- Y si cambiamos el formato de la salida?

E: 7 5 8

S: 8, 6, 9 ↵

- Idea: el 1er elem es el del formato diferente.

```
int x;
bool primer = true;
// Inv: x es un elemento de la secuencia o no hay más elementos
//     primer es cierto si no se ha tratado el primer elemento
//     de la secuencia, falso en caso contrario
while (cin >> x) {
    // tratar el elemento que acabamos de leer
    if (primer) {
        cout << x + 1;
        primer = false;
    } else cout << ', ' << x + 1;
}
cout << endl;
```

Tratamiento formato de salida

- Y si cambiamos el formato de la salida?

E: 7 5 8

S: 8, 6, 9 ↵

- Idea: el 1er elem es el del formato diferente.

```
int x;
bool primer = true;
// Inv: x es un elemento de la secuencia o no hay más elementos
//     primer es cierto si no se ha tratado el primer elemento
//     de la secuencia, falso en caso contrario
while (cin >> x) {
    // tratar el elemento que acabamos de leer
    if (primer) primer = false;
    else cout << ',';

    cout << x + 1;
}
cout << endl;
```