Lab 0: Resolviendo is_upperbound

Dados un arreglo de enteros “arr” y un valor entero “value”, debemos **determinar si el valor es cota superior del arreglo**. Dicho de otra manera, determinar si el valor es mayor o igual que todos los elementos del arreglo.

Al estilo Algoritmos I

Este problema se resuelve perfectamente con lo aprendido en Algoritmos I.

Especificación

Podemos especificar con pre y post condición:

```
Const arr : Array[0,N) of Int;
Const value : Int;
Var is_upper : Bool;
{ P: N ≥ 0 }
  S
{ Q: is_upper =⟨ ∀ i : 0 ≤ i < N : arr[i] ≤ value ⟩}
```

Derivación y programa resultado

Nos salteamos la derivación pero es **fácilmente realizable usando la técnica de reemplazo de constante por variable**.

El programa resultado es:

```
Const arr : Array[0,N) of Int;
Const value : Int;
Var is_upper : Bool;
Var i : Int;
{ P: N ≥ 0 }
  is_upper, i := True, 0;
  do i < N →
    is_upper, i := is_upper ∧ (arr[i] ≤ value), i + 1
  od
{ Q: is_upper =⟨ ∀ i : 0 ≤ i < N : arr[i] ≤ value ⟩}
```

Traducción a lenguaje C

La traducción más directa que se puede hacer es la siguiente:

```
int value;
int arr[];
unsigned int N;

bool is_upper = true;
int i = 0;
while (i < length) {
    is_upper = is_upper && (arr[i] <= value);
    i = i + 1;
}
```

También se puede usar un ciclo “for” en lugar de “while” (ejercicio 2).

Testing

Recomendamos probar muchos ejemplos, incluyendo sobre todo casos de borde (situaciones extremas). En la tabla de abajo se muestran posibles casos de test.

Para facilitar el testing, se puede programar una función que ejecuta todos los tests y chequea los resultados automáticamente (ejercicio 3).

arr	value	is_upper
{0, -1, 9, 4}	9	true
{0, -1, 9, 4}	**8**	**false**
{0, -1, 9, 4}	100	true
{9, -1, 0, 4}	8	false
{9}	9	true
{9}	100	true
{9}	8	false

Recordar que en C no hay arreglos vacíos, por lo que los arreglos de un elemento son casos de borde.

Ejercicios complementarios

1. Responder: ¿cuál es el invariante del ciclo?
2. Hacer versión con for en lugar de while.
3. **Programar una función que ejecute todos los casos de test y verifique los resultados.**
   Llamar a esta función desde la función `main()`.