

# Tarea Evaluativa

Abel Ponce González C211  
Richard Alejandro Matos Arderí C211

29 de marzo de 2024

## Ejercicio 18

---

### Descripción

Dada una lista  $L$  de  $n$  elementos  $a_1, a_2, \dots, a_n$  devolver el índice del mínimo elemento de  $L$ .

### Salida

Para la salida debe imprimir el mínimo elemento de  $L$ .  
Por ejemplo: para  $L = [4, 3, 5, 6]$  debería imprimir:

|   |   |
|---|---|
| 1 | 1 |
|---|---|

### Logisim

Se dispondrá en INPUT los datos de entrada a partir de la dirección 0. La entrada se estructura de la siguiente forma:

- $w_0$  :  $n$ (Tamaño de la lista  $L$ )
- $w_{1:n}$  :  $L$

### SASM

En la sección `.data` se deben definir los valores de entrada de la siguiente forma:

- $n$  : un número de tamaño `dd` que representa al tamaño de la lista  $L$
- $array$  : un array de números de tamaño `dd` que representa  $L$

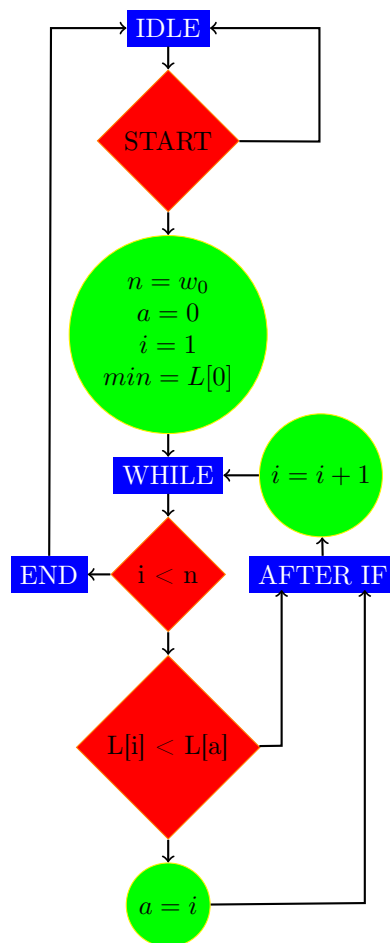
Por ejemplo, un posible encabezado podría ser:

|   |                                  |
|---|----------------------------------|
| 1 | <code>section .data</code>       |
| 2 | <code>n dd 4</code>              |
| 3 | <code>array dd 4, 3, 5, 6</code> |

## Seudocódigo

```
1      def Minimun(size, lista):
2          a = 0
3          i = 1
4          while i < size:
5              if lista[i] < lista[a]:
6                  a = i
7                  i = i + 1
8          return a
```

## DIAGRAMA



## Ejercicio 62

---

### Descripción

Determinar si una lista  $a_1, a_2, \dots, a_i$  está ordenada en orden creciente o decreciente.

### Salida

Para la salida debe imprimir la C si está ordenada en orden creciente o D lo está en orden decreciente.

Por ejemplo: para  $L = [3, 4, 6]$  debería imprimir:

```
1      C
```

## Logisim

Se dispondrá en INPUT los datos de entrada a partir de la dirección 0. La entrada se estructura de la siguiente forma:

- $w_0 : n$  (Tamaño de la lista  $L$ )
- $w_{1:n} : L$

## SASM

En la sección .data se deben definir los valores de entrada de la siguiente forma:

- $n$  : un número de tamaño  $dd$  que representa al tamaño de la lista  $L$
- $array$  : un array de números de tamaño  $dd$  que representa  $L$

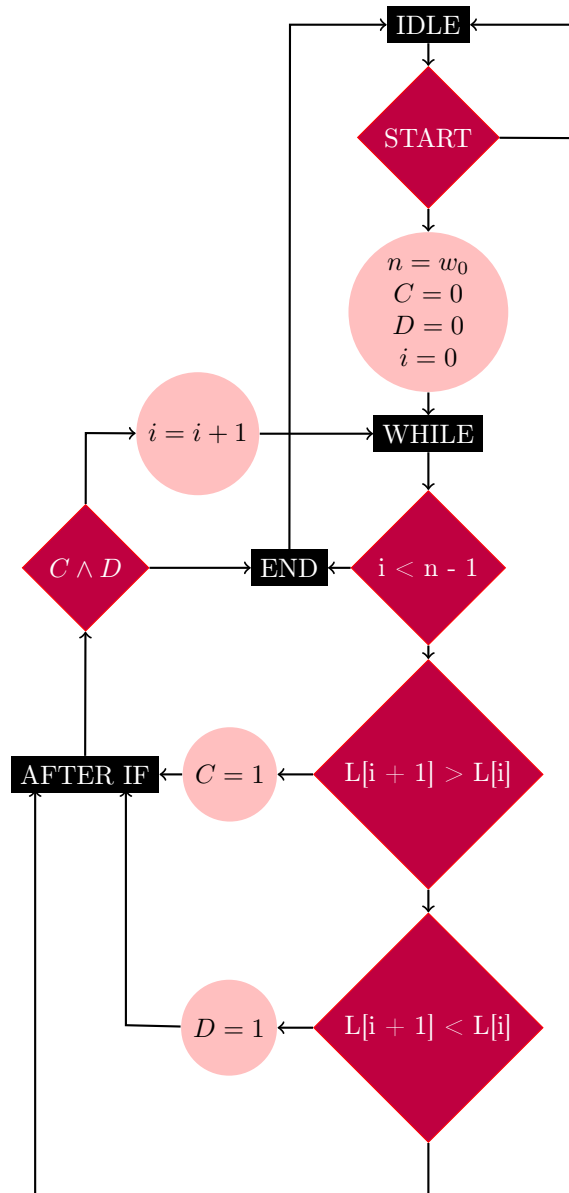
Por ejemplo, un posible encabezado podría ser:

```
1      section .data
2      n dd 3
3      array dd 3, 4, 6
```

## Seudocódigo

```
1      def Ordenacion (n, lista):
2
3          C = False
4          D = False
5          i = 0
6
7          while i < n - 1:
8
9              if lista[i + 1] > lista[i]: C = True
10
11              elif lista[i + 1] < lista[i]: D = True
12
13              if D & C : break
14
15              i = i + 1
16
17          if D & C : return "NO ORDENADA"
18          elif D : return "D"
19          else : return "C"
```

## DIAGRAMA



## Ejercicio 67

### Descripción

Dada una lista  $L$  devolver la lista  $L'$  que se obtiene al ordenar sus elementos, utilizando un método de ordenación  $O(n)$ . Se conoce que todos los elementos de la lista están en un intervalo desde  $a$  a  $b$  (ambos enteros); es decir, si el intervalo es de 3 a 6, la lista contiene a 3, 4, 5 y 6, pero no necesariamente ordenados. Note que el tamaño de la lista depende siempre del intervalo.

Dado una lista  $L$ , de tamaño  $n$ , que contiene todos los elementos de un intervalo desde algún  $a$  a un  $b$  (ambos desconocidos a priori), ordene la lista  $L$ .

### Salida

Para la salida debe imprimir la lista  $L$  ordenada de menor a mayor, separando cada elemento por un espacio en blanco.

Por ejemplo: para  $L = [4, 3, 5, 6]$  debería imprimir:

```
1      3 4 5 6
```

## Logisim

Se dispondrá en INPUT los datos de entrada a partir de la dirección 0. La entrada se estructura de la siguiente forma:

- $w_0 : n$  (Tamaño de la lista  $L$ )
- $w_{1:n} : L$

## SASM

En la sección .data se deben definir los valores de entrada de la siguiente forma:

- $n$  : un número de tamaño  $dd$  que representa al tamaño de la lista  $L$
- $array$  : un array de números de tamaño  $dd$  que representa  $L$

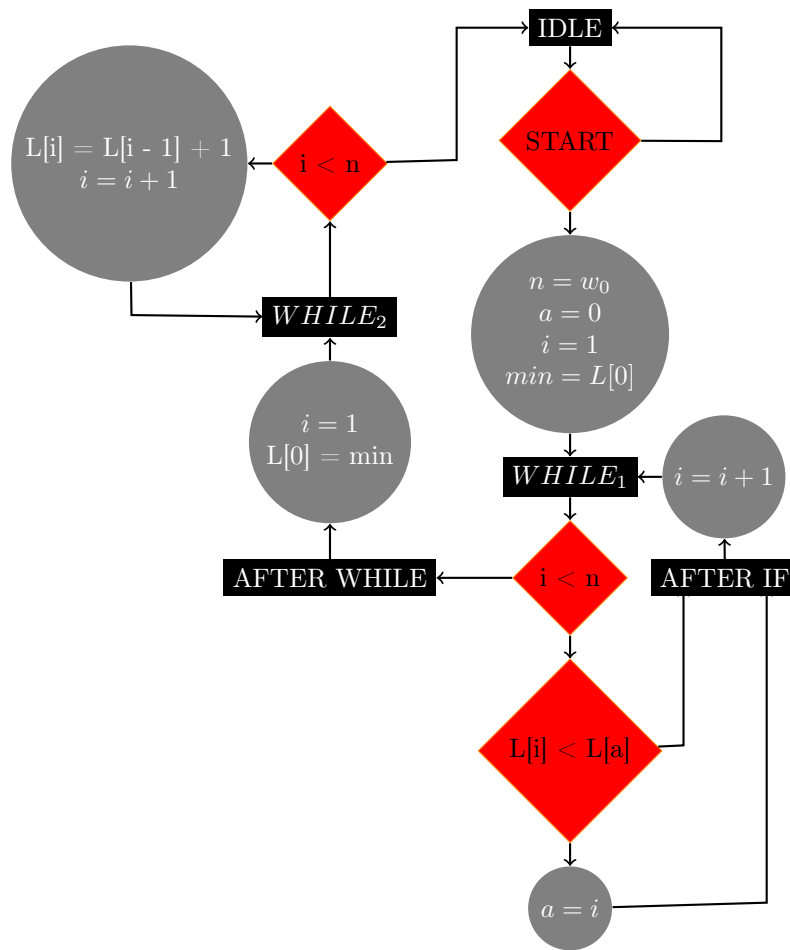
Por ejemplo, un posible encabezado podría ser:

```
1      section .data
2      n dd 4
3      array dd 4, 3, 5, 6
```

## Seudocódigo

```
1      def Order_On(n, L):
2
3          a = 0
4          i = 1
5
6          while i < n:
7
8              if L[i] < L[a]:
9                  a = i
10
11                 i = i + 1
12
13             i = 1
14             L[0] = L[a]
15
16             while i < n :
17                 L[i] = L[i - 1] + 1
18                 i = i + 1
19
20             return L
```

## DIAGRAMA



## Ejercicio 89

### Descripción

Dado un radio  $r$  de una circunferencia hallar el área de la misma.

### Salida

Para la salida debe imprimir el valor del área de la circunferencia en  $cm^2$ . Asuma que  $PI = 3$ ,  $PI$  es un entero.

Por ejemplo: para  $r = 2$  debería imprimir:

|   |    |
|---|----|
| 1 | 12 |
|---|----|

### Logisim

Se dispondrá en INPUT los datos de entrada a partir de la dirección 0. La entrada se estructura de la siguiente forma:

- $w_0 : r$

## SASM

En la sección `.data` se deben definir los valores de entrada de la siguiente forma:

- $r$  : un número de tamaño  $dw$  que representa  $r$

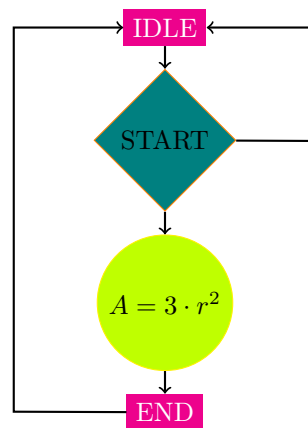
Por ejemplo, un posible encabezado podría ser:

```
1      section .data
2      r dw 2
```

## Seudocódigo

```
1      def Area (r):
2          return 3*r*r
```

### DIAGRAMA



### ASIGNACIONES

| $L_a$ | $r$ | $A$           |
|-------|-----|---------------|
| 0     | $r$ | 0             |
| 1     | $r$ | $3 \cdot r^2$ |