



Universidad de Oviedo



Escuela de  
Ingeniería  
Informática  
Universidad de Oviedo

---

# Trabajo en grupo

## Fase I

---

**Andrés Fernández-Junquera Fernández UO302806**

**Bruno Martín Rivera UO302144**

**Javier Ortín Rodenas UO299855**

**Mateo Rama García UO300710**

**Fundamentos de computadores y redes**

# Índice

<b>1. Primera parte</b>	<b>2</b>
1.1. PasswordControl() . . . . .	2
1.2. CountActiveBits() . . . . .	2
1.3. AsmBasedControl() . . . . .	2
1.4. ArrayMinMax() . . . . .	2
<b>2. Segunda parte</b>	<b>3</b>
2.1. Dirección de memoria IsValidAssembly . . . . .	3
2.2. Dirección de memoria PasswordControl . . . . .	3
2.3. Marco de pila ArrayMinMax . . . . .	3
2.4. Acceso de lectura ArrayMinMax . . . . .	3
<b>3. División del trabajo</b>	<b>4</b>

## 1. Primera parte

### 1.1. PasswordControl()

Parte de Andrés

### 1.2. CountActiveBits()

Esta función debe pedir dos números enteros sin signo. Posteriormente, debe contar el número de bits activos que hay en cada número entre la posición 5 y la 8, ambos inclusive. Finalmente, en caso de que el número de bits activos entre las posiciones 5 y 8 de los dos números no sea igual, la función imprimirá 'No coinciden' y llamará a la función exit().

- Entrada válida:  $a = 352$  y  $b = 448$
- Entrada no válida:  $a = 352$  y  $b = 0$

### 1.3. AsmBasedControl()

Esta función debe leer tres enteros y pasárselos a IsValidAssembly. Según la parametrización original del enunciado, esta segunda función debe comprobar que se cumplan las dos condiciones siguientes:

- El bit 8 del segundo número es igual al bit 5 del tercer número
- El valor de los 2 bits más bajos del primer número interpretados como binario natural es mayor que 11

Como la codificación de 11 en binario natural es 1011b, la segunda condición no puede ocurrir nunca en estas condiciones. Por tanto, escribimos la función para que tome los 4 bits más bajos del primer entero, los interprete como natural, y lo compare con 11.

Entrada válida:  $a = -3$ ,  $b = 403$ ,  $c = 56$

Entrada no válida:  $a = 1$ ,  $b = 7$ ,  $c = 5$

### 1.4. ArrayMinMax()

Parte de Bruno

## **2. Segunda parte**

### **2.1. Dirección de memoria IsValidAssembly**

[PLACEHOLDER, CAPTURA DE PANTALLA]

### **2.2. Dirección de memoria PasswordControl**

[PLACEHOLDER, CAPTURA DE PANTALLA]

### **2.3. Marco de pila ArrayMinMax**

[PLACEHOLDER, CAPTURA DE PANTALLA]

### **2.4. Acceso de lectura ArrayMinMax**

[PLACEHOLDER, CAPTURA DE PANTALLA]

### 3. División del trabajo

En primer lugar, dedicamos tiempo a leer el enunciado y repartir las tareas. Cada alumno eligió una función para programarla y responder las preguntas sobre ella de la segunda fase. [PLACEHOLDER ACABAR]