

Nombre: \_\_\_\_\_ NIU: \_\_\_\_\_

**PREGUNTA 4 (1,5 puntos)** Traducir al lenguaje ensamblador visto en las clases de problemas (RISC V) la siguiente sentencia del lenguaje C que compara dos números e identifica el mayor:

```
if (X > Y) Z = X  
else Z = Y
```

**;La dirección de la variable X en el registro R4**

LLA R4, X

**;La dirección de la variable Y en el registro R3**

LLA R3, Y

**;La dirección de la variable Z en el registro R5**

LLA R5, Z

**;El registro R1 contiene el valor de de la variable X**

LW R1, 0(R4)

**;El registro R2 contiene el valor de de la variable Y**

LW R2, 0(R3)

**;Comparación (X > Y) del bucle if**

BLE R1, R2, else

**;Z = X**

SW R1, 0(R5)

J fin

**else:**

**;Z = Y**

SW R2, 0(R5)

**fin:**

**PROBLEMA 5 (4 puntos)**

El código C mostrado a continuación, implementa una sencilla versión del algoritmo de ordenación de la burbuja (de menor a mayor). Traducir el código C situado en el rectángulo a lenguaje ensamblador del procesador usado en las clases de problemas (**RISC V**).

```
int temp;  
int fin;  
int i;
```

```
int a [10];
do {
    fin = 1;
    for (i = 0; i < 9; i++) {
        if (a[i] > a[i+1]) {
            fin = 0;
            temp = a[i];
            a[i] = a[i+1];
            a[i+1] = temp;
        }
    }
} while (fin != 0);
```

Hay que situar las instrucciones máquina en los recuadros situados a continuación de los comentarios identificados por ";"

```
;int temp;
;int fin;
;int i;
;int a [10];
```

*;La dirección inicial del array *a* en el registro R1*

*LLA R1,a*

*;El registro R2 contendrá el valor 9 (utilizado en la comparación del for)*

*LI R2,9*

```
;do {
;    fin = 1;
```

*inicio\_while:*

*;El registro R3 contiene directamente el valor de la variable fin*

*LI R3,1*

```
;    for (i = 0; i < 9; i++) {
```

*;El registro R4 contiene directamente el valor de la variable i (la primera iteración del for  
;valdrá 0)*

**LI R4,0**

**inicio\_for:**

*;Comparación *i* < 9 del for*

**BGE R4,R2, fuera\_for**

```
;        if (a[i] > a[i+1]) {
```

*;R5 almacena el contenido de a[i]*

```
LW R5,0(R1)
```

;R6 almacena el contenido de a[i+1]

```
LW R6,4(R1)
```

;Comparación a[i] > a[i+1] del if

```
BLE R5,R6, fuera_if
```

; fin = 0;

```
LI R3,0
```

; temp = a[i];

;El registro R7 contiene directamente el valor de la variable temp

```
MV R7, R5
```

; a[i] = a[i+1];

```
SW R6,0(R1)
```

; a[i+1] = temp;

```
SW R7,4(R1)
```

; }

fuera\_if:

; Dirección del siguiente elemento del array a , ya que no incremento respecto a i \*/

```
ADDI R1,R1,4
```

; i++ (del for)

```
ADDI R4,R4,1
```

; Volver al bucle for

```
J inicio_for
```

; }

fuera\_for:

; } while (fin != 0);

```
LI R8,0
```

```
BNE R3,R8,inicio_while
```

AYUDA: Sintaxis instrucciones ensamblador utilizado en problemas.

- LLA Rd, símbolo, LW/SW Rd, inm (Rm)
- LI Rd, imm
- MV Rd, Rm
- ADD / SUB / MUL/ DIV / SRA / SLL / SRL / AND / OR /XOR Rd, Rm, Rs

- **ADDI / SUBI / MULI / DIVI / SRAI / SLLI / SRLI / ANDI / ORI / XORI Rd, Rm, inm**
- **Enteros con signo: B{cond} etiqueta . Enteros sin signo: B{cond}U etiqueta**, donde **cond** puede ser: **EQ**: Igual, **NE**: No igual, **GE**: mayor o igual , **LT**: menor , **GT**: mayor , **LE**: mas pequeño o igual.
- **J etiqueta**
- **JAL Rd, etiqueta, RET**