

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 menor:

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

**;La dirección de la variable X en el registro R10**

LLA R10, X

**;La dirección de la variable Y en el registro R9**

LLA R9, Y

**;La dirección de la variable Z en el registro R11**

LLA R11, Z

**;El registro R7 contiene el valor de de la variable X**

LW R7, 0(R10)

**;El registro R8 contiene el valor de de la variable Y**

LW R8, 0(R9)

**;Comparación (X > Y) del bucle if**

BLE R7, R8, else

**;Z = X**

SW R7, 0(R11)

J fin

**else:**

**;Z = Y**

SW R8, 0(R11)

**fin:**

**PROBLEMA 5 (4 puntos)**

El código C mostrado a continuación, implementa una sencilla versión del algoritmo de ordenación de la burbuja (de mayor a menor). 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 c [20];
do {
    fin = 1;
    for (i = 0; i < 19; i++) {
        if (c[i] < c[i+1]) {
            fin = 0;
            temp = c[i];
            c[i] = c[i+1];
            c[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 c [20];
```

*;El registro R1 contendrá el valor 19 (utilizado en la comparación del for)*

LI R1,19

LLA R2,c

```
;do {
```

```
;    fin = 1;
```

inicio\_while:

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

LI R3,1

```
;    for (i = 0; i < 19; 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 < 19 del for*

BGE R4,R1, fuera\_for

```
;        if (c[i] < c[i+1]) {
```

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

```
LW R5,0(R2)
```

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

```
LW R6,4(R2)
```

;Comparación c[i] < c[i+1] del if

```
BGE R5,R6, fuera_if
```

; fin = 0;

```
LI R3,0
```

; temp = c[i];

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

```
MV R7, R5
```

; c[i] = c[i+1];

```
SW R6,0(R2)
```

; c[i+1] = temp;

```
SW R7,4(R2)
```

; }

fuera\_if:

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

```
ADDI R2,R2,4
```

; 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*, *imm*
- **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**