Cada respuesta incorrecta, ilegible o vacía no suma ni resta. **Algunas preguntas requieren archivos que deben formar parte de la entrega. No se corregirán preguntas a las que les falten los archivos indicados**. Al final del examen debes entregar los archivos que se enumeran en la última página.

---

1 ❑ Este ejercicio lo debes realizar empleando el simulador web de la CPU teórica que puedes encontrar en `http://www.atc.uniovi.es/tools/CPU/simulador.php`

(0,5 puntos) Inicializa manualmente (no por código) las direcciones de memoria 6110h a 6113h con los valores 1600h a 1603h, respectivamente. Asimismo inicializa manualmente la dirección de memoria 6114h con el valor 0.

Finalmente, configura estos registros con los valores indicados:

- `r0` con el valor 4.
- `r6` con el valor 3.
- `r3` con el valor 6110h.
- `r2` con el valor 6114h.

Realiza una captura de la ventana del simulador y guárdala en el archivo **`prog1-1.png`**. Esta captura debe formar parte de la entrega del examen.

(2 puntos) Las inicializaciones realizadas manualmente en el apartado anterior se corresponden con variables y constantes del siguiente programa que debes traducir, por lo que no deben repetirse en el código. **El código ensamblador resultante debe estar suficientemente comentado**.

```
// r3 = &array; r2 = &addval
int array[4] = {0x1600, 0x1601, 0x1602, 0x1603};
int addval = 0;

// r0 = 4; r6 = 3; r1 <-- i
for (unsigned int i = 0; i < 4; i++)
{
    if (i != 3)
        addval = addval + array[i];
}
```

Realiza una captura de la ventana del simulador completa con el código traducido sin ejecutar y guárdala en el archivo **`prog1-2.png`**. Esta captura debe formar parte de la entrega del examen.

(0,5 puntos) Simula el programa hasta el final y haz una captura de la ventana del simulador que muestre el valor de la variable `addval` al final del programa. La captura debe guardarse en el archivo **`prog1-3.png`** y formar parte de la entrega.

2 ❑ Se tiene el siguiente fragmento de código en lenguaje ensamblador del CT:

```
    movl r0, 61
    movh r0, 0
    movl r5, 39
    movh r5, 0
    movl r2, 36
    movh r2, 0
next:
    sub r3, r2, r0
    brnc next
    mov r3, [r4]
    dec r5
    sub r1, r5, r2
    brnz next
```

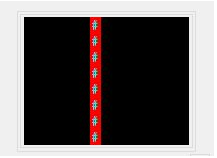
(0,5 puntos) Crea un archivo fuente ensamblador para el computador teórico que incluya el código anterior y se cargue a partir de la dirección de memoria 8500h. Compila el programa, cárgalo en memoria, y realiza el desensamblado del mismo usando la opción que proporciona el simulador del CT. Finalmente realiza una captura de pantalla del simulador en la que se vea tanto el desensamblado de memoria de todo el programa anterior como el valor del contador de programa y guárdalo en el archivo **`prog2-1.png`**. Este archivo debe formar parte de la entrega.

(1 punto) ¿Qué señales de control se activan en el paso 5 de la instrucción `mov r4, [r3]` la primera vez que se ejecuta? ¿Qué valor hay en el bus interno en ese mismo momento? Las respuestas deben proporcionarse en la hoja de respuestas escaneada. Además, debe proporcionarse una captura de todo el simulador del CT que debe guardarse con el nombre **`prog2-2.png`** y formar parte de la entrega.

Paso 5:

Valor en IB:

3 ❑ Se quiere hacer un programa que escriba en una columna de la pantalla un determinado carácter con unos atributos de presentación, de una manera similar a la mostrada en la figura. El primer carácter de la columna puede estar en una posición distinta a la mostrada. El carácter también puede ser distinto al mostrado.



El programa principal calcula la posición de la memoria y los atributos con los que se tiene que representar el carácter y realiza una llamada al procedimiento `putCharInScreen`. El procedimiento `putCharInScreen` recibe dos parámetros:

- la dirección de la pantalla en la que tiene que pintar el carácter
- el código ASCII y el atributo del carácter a pintar

El procedimiento hace que se muestre un carácter por pantalla, por lo que el programa principal debe llamarlo 8 veces. En el archivo `IOScreen.ens` se dispone de un esqueleto del programa.

Datos:

Columna:	12 (columnas numeradas de 0 a 14)
Carácter:	'#'
Color carácter:	Azul
Color del fondo:	Amarillo

(2 puntos) Completa el programa y el procedimiento teniendo en cuenta que `Addr_1stChar` es la dirección en la que se dibuja el primer carácter de la primera fila. Guárdalo en el archivo **`IOScreen.ens`**. Incluye **`IOScreen.ens`** entre los archivos a entregar.

(0,5 puntos) Configura el simulador conectando un periférico pantalla y asegurándote que el Computador Teórico dispone de la máxima cantidad posible de memoria RAM compatible con el periférico pantalla. Ten en cuenta que `Addr_1stChar` es la dirección en la que se dibuja el primer carácter.

(0,5 puntos) Compila el programa `IOScreen.ens` y genera `IOScreen.eje`. Cárgalo en el simulador y ejecútalo hasta que se pinten 4 caracteres en pantalla. Salva el estado del simulador en ese momento en el archivo `IOScreen.sim`. Adjunta **`IOScreen.eje`** e **`IOScreen.sim`** entre los archivos a entregar.

A

4 ❑ En el archivo `Interrupt.sim` se dispone de la configuración del simulador del CT con dos dispositivos conectados. Se quiere hacer un programa que cuando se genere una interrupción en el dispositivo Switch2, se lea el estado de los interruptores del dispositivo y se muestren en los leds del dispositivo Switch1. El programa constará de dos partes:

- La rutina de servicio `switch2Routine` que leerá del dispositivo Switch2 el estado de los interruptores y los mostrará en los leds del dispositivo Switch1.
- El programa principal que se limitará a instalar la rutina, activar las interrupciones y esperar en un bucle infinito.

En el archivo `Interrupt.ens` se dispone de un esqueleto del programa.

(2 puntos) Completa el código del programa y la rutina de servicio en el archivo `Interrupt.ens`. Incluye **`Interrupt.ens`** entre los archivos a entregar.

(0,5 puntos) Compila el programa `Interrupt.ens` y genera `Interrupt.eje`. Cárgalo en el simulador y ejecútalo hasta que se instale la rutina de interrupción. Salva el estado del simulador en ese momento en el archivo `SolucionInt.sim`. Adjunta **`Interrupt.eje`** y **`SolucionInt.sim`** entre los archivo a entregar.

Al final del examen debes entregar la hoja de respuestas escaneada, firmada y con documento que te identifique, incluyendo la respuesta al apartado del problema 2. Además debes incluir los siguientes archivos:

- **`prog1-1.png`**, **`prog1-2.png`** y **`prog1-3.png`**.
- **`prog2-1.png`** y **`prog2-2.png`**.
- **`IOScreen.ens`**, **`IOScreen.eje`** e **`IOScreen.sim`**.
- **`Interrupt.ens`**, **`Interrupt.eje`** e **`SolucionInt.sim`**.