Apellidos: ..... Nombre: .....

DNI: .....

1. ¿Cuál es la función del procesador de un computador?

   El procesador es el encargado de ejecutar las instrucciones almacenadas en la memoria principal y de generar las señales de control que rigen el funcionamiento del ordenador.

2. Comenta brevemente cómo interactúan los distintos componentes de un computador para ejecutar la instrucción «`bl subrutina`».

   En primer lugar, como con todas las instrucciones, el procesador activa las señales de control necesarias y envía el contenido del contador de programa a la memoria. Esta le devuelve la instrucción a ejecutar y el procesador la almacena y decodifica.

   Posteriormente, el procesador lee el contenido del contador de programa, debidamente incrementado tras leer la instrucción, y el valor constante del desplazamiento contenido en la propia instrucción. Mediante su ruta de datos, desplaza esta constante y la suma así modificada, al contenido del contador de programa. A continuación, escribe el valor leído del contador de programa, que corresponde con el de la siguiente instrucción, en el registro `lr` y el resultado de la suma, en el contador de programa, verificando así el salto a la subrutina.

   Como se ha dicho, durante alguna de las etapas anteriores, el procesador ha actualizado el contenido del contador de programa para que guarde la dirección de la siguiente instrucción. En este caso, este valor se almacenará en `lr`, pues el `pc` se modifica con la dirección de la subrutina.

3. Codifica en binario y en hexadecimal la instrucción «**`strh r0, [r7, #44]`**» sabiendo que el formato de instrucción utilizado para codificar dicha instrucción es el siguiente:

L Load/Store: 1, cargar; 0, almacenar.

Offset5 Dato inmediato.

Rb Registro base.

Rd Registro fuente/destino.

Veamos los campos en binario, de izquierda a derecha:

- 1000 es fijo y nos indica de qué tipo de instrucción se trata.
- El bit L valdrá 0 pues se trata de una instrucción de almacenamiento.
- El Offset5 será el valor 44, que se codifica dividido entre 2 pues el tamaño del acceso es media palabra. Por lo tanto, se codifica 22 en 5 bits, que en binario es 10110.
- El registro base, Rb, es r7, 111.
- El registro destino, Rd, es r0, 000.

Juntando los campos se tiene:

Que en hexadecimal es 0x85B8.

4. Identifica y describe el modo de direccionamiento de cada uno de los operandos de la instrucción «**`add r0, #47`**».

   La instrucción tiene dos modos de direccionamiento, dado que el primer operando, `r0`, es fuente y destino. El segundo operando es fuente y se expresa como `#47`.

   El primer modo de direccionamiento es directo a registro. Cuando se considera r0 como fuente, se indica el registro cuyo valor se va a leer para efectuar la suma. Cuando se considera como destino, indica el registro dentro del que se va a escribir el resultado.

   El segundo es el modo de direccionamiento inmediato o literal. En este caso, el valor 47 está contenido dentro de la propia instrucción, en el campo correspondiente.

5. Dado el siguiente programa en ensamblador Thumb de ARM, **describe su funcionamiento comentando cada una de sus líneas** y anota a continuación el contenido final de los registros `r0`, `r1`, `r2` y `r6` y el de la palabra almacenada en la dirección de memoria `0x2007 0004`.

```
1         .data
2
3  dato:  .word 52063
4  res:   .space 4
5  vals:  .word 3, 117, 20850, 187921, 433256, 1000000
6
7         .text
8
9  main:  ldr r7, =dato          @ Dirección del valor a comparar
10        ldr r6, =vals          @ Dirección inicial del vector
11        mov r0, #0             @ Contador de posición
12
13        ldr r1, [r7]           @ En r1 se almacena dato
14 buc:   ldr r2, [r6]           @ Se lee un elemento del vector
15        cmp r2, r1             @ y se compara con dato
16        bhi fin                @ Si es mayor -sin signo-, se termina
17        add r0, r0, #1         @ Si no, se incrementa la posición y se calcula
18        add r6, r6, #4         @  la dir. del siguiente elemento en el vector
19        b   buc
20 fin:   str r0, [r7, #4]       @ En res se almacena el contador de posición
21        wfi
22
23        .end
```

El programa propuesto compara el valor del entero almacenado en dato con los del vector vals, que están ordenados de manera creciente, e indica la posición dentro del vector, del primer elemento mayor que aquél. Las comparaciones se realizan sin signo. Como los elementos del vector están ordenados, en cuanto se encuentra uno mayor al dato, el programa termina.

`r0`	3
`r1`	52063
`r2`	187921
`r6`	0x20070014
`0x2007 0004`	3