

**Informática: Arquitectura de
Ordenadores
Curso 2020-2021**

PRÁCTICA 6

Esta práctica incluye tres ejercicios de iniciación a los subprogramas en ARM-Thumb. No está permitido en esta práctica el uso de la pila.

1.- Realice un subprograma, `cuadrados`, que dada una matriz NxN de enteros de 16 bits, eleve al cuadrado los elementos de su diagonal. El subprograma recibirá en `r0` la dirección de comienzo de la matriz original y en `r1` su dimensión (N). La matriz estará serializada por filas.

Ejemplo de invocación:

```

                .data
matriz:         .hword 1, 2, 3    @ matriz 3x3
                .hword 4, 5, 6
                .hword 7, 8, 9

                .text
main:           ldr r0, =matriz @ direccion de comienzo de matriz
                mov r1, #3       @ dimension 3
                bl  cuadrados
                wfi
```

matriz resultante: 1, 2, 3, 4, 25, 6, 7, 8, 81 = 0x01, 0x0002, 0x0003, 0x0004, 0x0019, 0x0006, 0x0007, 0x0008, 0x0051

2.- Realice un subprograma, a partir de la posición etiquetada como `search` que reciba en `r0` y `r1` la dirección de dos cadenas de caracteres ASCII finalizadas con un 0. El subprograma devolverá, en `r3`, un 1 si la cadena a la que apunta `r0` se encuentra a partir de la dirección contenida en `r1`. En caso contrario devolverá un 0. Si alguna de las dos cadenas está vacía devolverá un -1

En el caso:

```
cadena0:  asciz: "ama"
cadena1:  asciz      "amame mi ama"
```

El resultado sería 1.

En el caso:

```
cadena0:  asciz    "ama"
cadena1:  asciz    "me amas"
```

El resultado sería 0

Utilizando el subprograma anterior realice un programa que cuente el número de apariciones de una secuencia de caracteres en otra de mayor o igual tamaño. La cadena a buscar (que nunca estará vacía) está situada a partir de la posición de memoria etiquetada como 'findme' y su finalización la marca un 0. La cadena sobre la que se realizará la búsqueda se encuentra en la posición de memoria etiquetada como 'searchin' y su finalización está marcada por un cero. El programa devolverá en r0 el número de apariciones.

```
        .data
findme:  asciz    "ama"
searchin: asciz    " amama  ama"
```

Con los datos del ejemplo anterior en r0 deberá quedar un 3

3.- Realice un subprograma, a partir de la posición etiquetada como `menor`, que reciba la dirección de comienzo de una secuencia de enteros de 32 bits en r0 y devuelva, en el propio r0, la posición de memoria en la que se encuentra el menor de los enteros. La secuencia terminará con un 0. Utilice este subprograma para realizar un programa que, dada una secuencia de enteros de 32 bits (terminada en 0), la ordene de menor a mayor.

Ejemplo:

```
        .data
nums:   .word 5, 4, 3, 1, 2, 2, -1, -2, 0
```

El programa debiera dejar la secuencia: -2, -1, 1, 2, 2, 3, 4, 5, 0