

Laboratorio 4: Arquitectura y Organización de Computadores

Profesores: Mauricio Solar y Viktor Tapia

Ayudantes: Joaquín Montes - Benjamín López - Mauricio Cortés - Javier Rojas

1. Reglas generales

Para esta tarea se deberá utilizar el software QtARMSim¹ para escribir un programa que resuelva los problemas propuestos en la sección de Enunciado. El programa debe utilizar el lenguaje ARM Assembly y debe ser ejecutable en QtARMSim sin necesidad de plugins adicionales.

2. Enunciado

Se le pide crear un programa en QtARMSIM que reciba como primer parámetro un número entero entre 1 y 3, y dependiendo de este número se ejecute una de las siguientes funciones.

2.1 Función 1: Verificar Anagrama

Esta función debe recibir 2 string y retornar 1 en caso de que los string sean anagramas entre sí, 0 en caso contrario. Para ello se les recomienda crear las siguientes 3 sub rutinas.

- La subrutina sub1 recibe dos strings y retorna 1 si ambos strings son anagramas² entre si, y 0 en caso contrario. Para esto invocará las subrutinas sub2 y sub3.
- La subrutina sub2 recibe un string y cuenta cuántas veces se repite cada carácter del string.
- La subrutina sub3 recibe dos vectores y retorna un 1 si contienen lo mismo, y un 0 en caso contrario.

2.2 Función 2: Función Recursiva

Se debe traducir el código en Python, mostrado en la Figura 1, a código ARM.

```
def recursive(n, k):  
    if k > n:  
        return 0  
    elif n == k or k == 0:  
        return 1  
    else:  
        return recursive(n-1, k) + recursive(n-1, k-1)
```

Figura 1: Código de una función recursiva en Python

2.3 Función 3: Múltiplos de 2

Se debe implementar un programa que busque los valores pares (múltiplos de 2) de un vector, y retorne la cantidad de pares encontrados y los pares en cuestión. Para esto define que la función debe recibir un vector y la dimensión o largo del vector.

¹<https://pypi.org/project/qtarmsim/>

²Dos palabras son anagramas si y solo si contienen exactamente las mismas letras, pero sin importar el orden. Por ejemplo, las palabras amor y mora son anagramas, pero mar y rama no lo son.

3. Entrada y salida de datos

La entrada y salida de datos se hará a través de las interfaces stdin (.data) y stdout (Pantalla LCD) de QtARMSim. Los datos se recibirán en dos pasos, primero la operación a realizar y luego los valores para dicha operación.

Todos los strings tendrán un largo máximo de 10 caracteres, y todos los valores numéricos estarán entre 0 y 25, incluyendo ambos límites.

Valor	Descripción	Parámetros
1	Análisis de anagramas	2 strings
2	Recursividad	Valores para n y k
3	Búsqueda de Pares	La dimensión de un vector y el vector.

4. Datos de ejemplo

A continuación se presenta una serie de entradas y sus respectivas salidas tras ser procesadas por el programa.

Input	Output
1 amor mora	1
1 mar rama	0
2 n = 5 k = 2	10
2 n = 10 k = 10	1
2 n = 5 k = 2	0
3 largo = 5 array = [1,2,3,4,5]	2 2 4
3 largo = 3 array = [1,3,5]	0

5. Consideraciones

- Se deberá trabajar de a pares. La solución al laboratorio debe entregar en Aula a más tardar el día 25 de Junio a las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de tareas.
- La tarea debe realizarse usando el lenguaje ARM Assembly y el simulador QtARMSim.
- La tarea se debe entregar vía Aula en un solo archivo comprimido en formato .zip de nombre LAB4_ROL1_ROL2.zip que incluya los siguientes archivos:
 - Un solo archivo README.txt con el nombre y ROL USM de los estudiantes, además de cualquier aclaración que sea necesaria.
 - Un solo archivo .s que contenga todo el código en ARM Assembly para la ejecución de la tarea.
- **Todas las preguntas respecto a la tarea deben hacerse a través del foro de consultas en Aula. No se responderán dudas durante las 48 horas previas a la entrega.**
- **La fecha límite de entrega de la tarea es el 25 de Junio a las 23:59 horas.**