

Laboratorio 11

Competencias a desarrollar

Diseñar programas en el lenguaje ARM que utilicen el manejo de cadenas, arreglos y números en punto flotante en el lenguaje ensamblador ARM.

Instrucciones

1. Realice los siguientes programas de forma individual. **La mínima evidencia de copia entre estudiantes conlleva un 0 (cero) como nota.**
2. Entregue los archivos “.s” en la tarea de *Canvas*.
3. Utilice comentarios dentro del código, para separar las diferentes áreas del programa e indicar lo que hace cada paso.
4. Pruebe el programa varias veces, con diferentes valores en la localidad de memoria.

Programa 1

1. Cree un programa con el nombre **lab12programa1.s**
2. Recuerde poner un encabezado, definir el área de datos y el área de código. Además, defina el main y recuerde agregar las instrucciones para salir correctamente al sistema operativo al final del programa.
3. En la sección de datos defina un arreglo con el siguiente contenido:
`arreglo .float 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0`

NOTA: recuerde que para direccionar al elemento .float, previamente debe crear una etiqueta para el .word asociado, i.e.: `dir_arreglo: .word arreglo`

4. Ingrese los 10 datos del arreglo desde el teclado, verificando que se ingresen números.
5. Defina otro arreglo del mismo tamaño que el anterior, este segundo arreglo será utilizado para almacenar el resultado.
6. Tome el arreglo con los datos y almacénelo en el segundo arreglo resultado en orden inverso (invierta el orden del arreglo). Utilice el modo de direccionamiento pre-indexado para recorrer el arreglo con los datos y el modo post-indexado para almacenar el resultado.

Despliegue en pantalla los dos arreglos de la siguiente forma (por ejemplo, si los valores ingresados fueron 15.1, 20.2, 30.4, 40.7, 50.3, 60.9, 70.6, 80.7, 90.1, 100.5):

Original
15.1
20.2
30.4
40.7
50.3
60.9
70.6

80.7
90.1
100.5
Invertido
15.1
20.2
30.4
40.7
50.3
60.9
70.6
80.7
90.1
100.5

7. Realice la suma de todos los elementos del arreglo original y despliegue el resultado en pantalla.

Programa 2



1. Cree un programa con el nombre **lab12programa2.s**
2. Recuerde poner un encabezado, definir el área de datos y el área de código. Además, defina el main y recuerde agregar las instrucciones para salir correctamente al sistema operativo al final del programa.
3. Defina una cadena vacía.
 - a. La cadena vacía debe ir al final de su sección de datos. No defina ninguna variable después de la cadena resultado.
 - b. Recuerde que para definir una cadena vacía se utiliza la directiva **.asciz**, de la siguiente forma:
cadena: .asciz " "
4. Permita que el usuario ingrese su nombre **alternando caracteres en minúsculas y mayúsculas** desde el teclado y almacénelo en la cadena en memoria que definió en el inciso anterior. **OJO: EL NOMBRE NO DEBE TENER ESPACIOS EN BLANCO, LA FUNCIÓN DE C (SCANF) LO INTERPRETA COMO FIN DE CADENA**
 - a. **Nota:** el usuario no podrá ingresar más de 10 caracteres. Ejemplo:
Por favor ingrese su nombre (max. 10 caracteres): _
5. Por medio de OPERACIONES LÓGICAS convierta la cadena: **los caracteres que originalmente se ingresaron en minúsculas deberán convertirse a mayúsculas, y los que se ingresaron en mayúsculas deberán convertirse a minúsculas**. Almacene el resultado en otra cadena, similar a la definida en el paso 3.
 - a. **Importante:** averigüe el rango del código ASCII de las letras mayúsculas y el de las minúsculas. Notará que la diferencia entre una mayúscula y una minúscula es un bit. Por ejemplo: 'a' es 0x61 y 'A' es 0x41.
 - b. **Nota:** después de convertir la cadena no olvide agregar el carácter **NULL** (valor cero) al final.

6. Despliegue en pantalla el resultado del nombre. Por ejemplo:

Por favor ingrese su nombre (max. 10 caracteres): Luis

Su nombre en mayusculas: LUIS

Evaluación

Criterios de evaluación	Nivel 3 Experto 	Nivel 2 Aprendiz 	Nivel 1 Novato 
Funcionamiento del Programa del primer programa del laboratorio 30%	El programa funciona con todos sus requerimientos: ingreso de datos, despliegue de resultados correctos y salida correcta al sistema operativo 30%	El programa funciona en al menos el 60% de sus requerimientos de funcionamiento. 18%	El programa funciona en menos del 50% de sus requerimientos. 4%
Funcionamiento del segundo programa del laboratorio 35%	El programa funciona con todos sus requerimientos: ingreso de datos, despliegue de resultados correctos y salida correcta al sistema operativo. 35%	El programa funciona en al menos el 60% de sus requerimientos de funcionamiento. 15%	El programa funciona en menos del 50% de sus requerimientos. 5%
Programación defensiva de ambos programas 10%	Los programas tienen muy buena programación defensiva en todos los ingresos de datos, y proporcionan mensajes oportunos ante situaciones inesperadas. 10%	Los programas tienen programación defensiva en la mayoría de los ingresos de datos, y proporciona algunos mensajes oportunos ante situaciones inesperadas. 5%	Los programas tienen muy poca o ninguna programación defensiva en los ingresos de datos, y proporciona pocos o ningún mensaje oportuno ante situaciones inesperadas. 0%
Documentación de ambos programas 5%	La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes. Los nombres de las variables son significativos. 5%	Falta documentación en el encabezado o en bloques de código Los nombres de las variables son medianamente significativos. 2.5%	Falta gran parte de la documentación del código Los nombres de las variables no expresan ningún significado. 0%
Orden de ambos programas 10%	La presentación del programa es muy clara y ordenada, y utiliza una tabulación adecuada. 10%	La presentación del programa es regularmente clara y ordenada. La tabulación es aceptable. 8%	La presentación del programa es confusa y desordenada. No hay tabulación de las instrucciones. 0%
Reflexión 10%	VER RÚBRICA DE REFLEXIÓN EN CANVAS		