

Guía de ejercicios # 10: Q6: Arreglos

Organización de computadoras

UNQ

Procesamiento de arreglos

1. Responder:

- ¿Qué es un arreglo? y ¿por qué son necesarios para programar?
- ¿Cómo se implementa, en bajo nivel, un arreglo en Q, es decir, dónde están sus elementos? y ¿qué característica tiene?
- ¿Qué se necesita para recorrer los elementos de un arreglo?
- ¿Qué modos de direccionamientos se utilizan para procesar los datos de un arreglo?
- ¿Cómo se conoce la posición en la que está el elemento que se está procesando?

2. Considerando el siguiente mapa de memoria y el valor de los registros:

	...
0000	0001
0001	0004
0002	0000
0003	0002
0004	0001
	...

Registro	Valor almacenado
R0	0x0001
R1	0x0003

Indicar el **efecto** para cada instrucción:

- MOV [R0], [[0x0003]]
- MOV [[0x0003]], [R0]
- MOV [[0x0003]], [[0x0001]]
- MOV [[0x0001]], [R1]

3. Definir una rutina que, dado un arreglo de 4 valores en *BSS*(16), retorne la suma de sus elementos. El arreglo se encuentra a partir de la celda con dirección *ADDD*.

- Dar un ejemplo de un posible escenario (caso) con los datos iniciales y el resultado esperado
- Documentar la rutina
- Definir la rutina que cumpla con la documentación previa

4. Se solicita documentar y definir la rutina **cantElementos** que, a partir de un arreglo de valores en *BSS*(16), retorne su cantidad de elementos. El arreglo se encuentra a partir de la dirección indicada por *R0*, y finaliza con el primer elemento cuyo valor es 0.

5. Definir la rutina **copiarArreglo** en función de su documentación:

copiarArreglo	
Requiere	un arreglo de cadenas a partir de la celda con dirección 8400 y que finalice con el primer elemento cuya cadena es 0xFFFF
Retorna	una copia del arreglo original a partir de la celda con dirección 9400
Modifica	COMPLETAR

6. Documentar y definir la rutina **aplicarAbsolute** que a partir de un arreglo de valores en *SM*(16), retorne dicho arreglo con sus elementos en valor absoluto. El arreglo se encuentra a partir de la celda con dirección 0x4486, y su longitud está en la celda con dirección 0x4485. **Nota:** usar la rutina **absolute** de la práctica anterior.

7. ¿Qué se debe modificar en la rutina **aplicarAbsolute** si el parámetro de la dirección inicial del arreglo se recibe en *R0*?

8. En una fábrica de ventanas se codifican los pedidos en cadenas de 16 bits, y para el caso de que la ventana tenga vidrio de seguridad o esmerilado, es necesario usar un embalaje distinto, por lo que se considera que el pedido es **Premium**. Al resto de los pedidos se los denomina **Básicos**.

El pedido se denomina **dataPedido** y cuenta con el siguiente formato:

RRRRRRR(5b)	S(1b)	CCCCC(5b)	E(1b)	PP(4b)
-------------	-------	-----------	-------	--------

Donde: **R** se utiliza como relleno, **S** para indicar si el vidrio es de seguridad o no, **C** para el código del pedido, **E** para indicar si el vidrio es esmerilado o no, y **P** para el precio.

- Escribir, en formato **dataPedido**, 4 ejemplos, que representen las combinaciones de pedidos posibles (premium y básico).
- Documentar y definir la rutina **esPremium**, que determine si un pedido dado en *R5* es premium o no. Retornar en *R7* un 1 en caso afirmativo, o un 0 en caso contrario.

Responder: ¿qué condición de corte hace que la rutina sea más eficiente?

(c) Dada la siguiente documentación:

cantBasicos	
Requiere	un arreglo de pedidos a partir de la celda con dirección 9ED1, y cuya longitud está en en la celda con dirección 9ED0
Retorna	en R6 la cantidad de pedidos que son básicos.
Modifica	COMPLETAR

- Dibujar un arreglo de pedidos a modo de ejemplo (mapa de memoria).
- Definir la rutina `cantBasicos` y completar el campo `Modifica`.
- Responder:** ¿qué salto se debe utilizar para no duplicar código? ¿Cuál es el parámetro de la rutina `esPremium` dentro de `cantBasicos`?

(d) Definir una rutina de test que pruebe el funcionamiento de la rutina `cantBasicos`, para el caso dado en el ejemplo del punto anterior.

rutina: `MOV R0, [R1]`
`ADD [[0x1810]], R0`
`RET`

Sabiendo que `R1 = 2234`, `[2234] = 000A`, `[1810] = 2020`, y `[2020] = 0008`.

- Ensamblar la rutina y alojarla en memoria a partir de la celda con dirección `1983`
- Simular la ejecución de la rutina, teniendo en cuenta el ciclo de ejecución de cada instrucción.

Bajo nivel: arquitectura

9. Para cada instrucción dada a continuación:

- Indicar el valor de los registros de uso específico luego de la búsqueda de la instrucción.
- Indicar a qué celdas se acceden en cada etapa del ciclo de ejecución, en base a la información dada en cada una (etapa y dirección). Indicar el efecto final.

Nota: cada instrucción es independientes entre sí.

- 1. `MOV R1, [[0x0000]]` ensamblada a partir de la dirección `B154` y sabiendo que en la celda con dirección `0000` se encuentra el dato `077A` y en la celda con dirección `077A` el dato `9066`.
- 2. `MOV [R2], [R1]`, ensamblada a partir de la dirección `EE54` y sabiendo que `R1` contiene el dato `F57A`, `R2` el dato `C32F` y `[F57A] = 1986`.
- 3. `MOV [[0x0000]], [[0x0001]]`, ensamblada a partir de la dirección `C154` y sabiendo que en las celdas con dirección `0000` y `0001` se encuentran los datos `0A44` y `B312` respectivamente. Donde `[B312] = 0002`.
- 4. `ADD [R2], [R1]`, ensamblada a partir de la dirección `1234` y sabiendo que `R1` contiene el dato `157A` y `R2` el dato `2AAF`. Donde `[157A] = 0x000A`, y `[2AAF] = 0x0004`.
- 5. `ADD [[0x0000]], [[0x0001]]`, ensamblada a partir de la dirección `4321` y sabiendo que en las celdas con dirección `0000` y `0001` se encuentran los datos `0B54` y `E012` respectivamente. Donde `[0B54] = 0x0008`, y `[E012] = 0x0003`.