

Guía de ejercicios # 10: Iteraciones, arreglos y recorridos

Organización de Computadoras 2018

UNQ

Objetivos

Que el estudiante pueda:

- Entender qué es una iteración y cómo las hacemos en Q
- Comprender el concepto de arreglo
- Entender la necesidad de un modo indirecto
- Rescatar/generalizar la estructura de los programas que recorren arreglos

Los ejercicios marcados con ★ forman un conjunto minimal para comprender los temas tratados en esta práctica.

Para resolver esta práctica se aconseja consultar los apuntes de la materia *Iteraciones y Arreglos*, disponible en <http://orga.blog.unq.edu.ar/descargas/>.

1 Repeticiones

1. Escribir **y documentar** una rutina que calcule la multiplicación $R5 * R6$ sin usar la instrucción MUL. Consejo: $A * B$ es sumar A unas $B - veces$, por ejemplo $8 * 3$ es $8 + 8 + 8$.
2. Escribir **y documentar** una rutina que calcule el la división entera $R0 \% R1$ sin usar la instrucción DIV.
3. ★ Escribir **y documentar** una rutina que cuente la cantidad de dígitos 1 en la cadena que está en R6.
4. ★ Escribir **y documentar** una rutina que calcule el factorial del valor almacenado en R5. dicho valor está representado en $BSS()$. El factorial de un número n es el producto de los números enteros positivos desde el 1 hasta n . Por ejemplo, el factorial de 5 es:

$$5! = 1 * 2 * 3 * 4 * 5$$

2 Modos indirectos

Para los siguientes cuadros de actividad de buses asumir lo siguiente:

- Las instrucciones están ensambladas a partir de la celda 9060.
- El siguiente estado parcial de registros y memoria.

			...
		0000	A000
		0001	A003
	
R1	A004	A000	0002
R2	A002	A001	9004
		A002	0003
		A003	0005
		A004	0001
			...

5. Considerar la instrucción ADD $[[0000]], R1$
 - (a) Ensamblar la instrucción.
 - (b) Completar el siguiente cuadro de uso de buses.

Etapa	Bus de control	Bus de dir.	Bus de datos

6. Considerar la instrucción MOV $[[0000]], R1$
 - (a) Ensamblar la instrucción.
 - (b) Completar el siguiente cuadro de uso de buses.

Etapa	Bus de control	Bus de dir.	Bus de datos

7. ★ Ensamblar el siguiente programa, ubicándolo a partir de la celda 0FF0

```
MOV R1, [[0000]]
volver: CMP [R2], 0x0000
```

8. ★ Completar la siguiente tabla de accesos para el programa anterior.

Instrucción	B.Inst.	B.Op.	Alm.Op.

9. Ensamblar el siguiente programa, ubicándolo a partir de la celda 0FF0

```
MOV R1, [[0x0000]]
MOV [R2], [R1]
MOV [[0x0000]], [[0x0001]]
ADD [R2], [R1]
ADD [[0x0000]], [[0x0001]]
```

10. Completar la siguiente tabla de accesos para el programa anterior.

Instrucción	B.Inst.	B.Op.	Alm.Op.

11. ★ Considerando el siguiente mapa de memoria y valor de los registros. ¿Qué registro o posición de memoria se termina modificando en cada instrucción? ¿Cuál es el valor que queda almacenado?

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

Registro	Valor almacenado
R0	0x0001
R1	0x0003

Por ejemplo, para la instrucción:

```
MOV [R1], [[0x0002]]
```

La celda que se va a modificar es la 0x0003. El valor que queda almacenado es 0x0001. (vamos a la dirección 0x0002 donde su valor es 0x0000 y en esa celda se encuentra el valor 0x0001).

- (a) MOV [R0], [[0x0003]]
- (b) MOV [[0x0003]], [R0]
- (c) MOV [[0x0003]], [[0x0001]]
- (d) MOV [[0x0001]], [R1]

3 Arreglos

12. Dado un arreglo que comienza en la celda 0x3333 y que tiene 5 elementos en *BSS*(16), se pide escribir una rutina que sume los valores del arreglo.
13. Dado un arreglo que comienza en la dirección que se encuentra en R0 (es decir R0 tiene la dirección del primer elemento del arreglo y [R0] el valor de dicho elemento), y que termina con el primer elemento cuyo valor es 0x0000; se pide escribir una rutina que cuente la cantidad de elementos que tiene el arreglo.
14. ★ Escribir una rutina `cantidadNrosPares` que cuenta cuantos números de un arreglo son pares. El arreglo comienza en la celda 0x4486 y la longitud del arreglo está en la celda 0x4485.
15. ★ Escribir una rutina `cantidadNrosQuintoBit1` que cuenta cuantos números de un arreglo tienen el quinto bit en 1. El arreglo comienza en la celda 0x4486 y la longitud del arreglo está en la celda 0x4485.
16. (a) Implemente la siguiente rutina a partir de su documentación

```

;-----maximoValorEnArregloABAB
; REQUIERE Un arreglo que comienza en la celda ABAB
;           Los valores ocupan 1 celda cada uno.
;           El tamaño del arreglo está en R7.
; MODIFICA ??
; RETORNA en R6 el valor máximo del arreglo
;-----

```

 - (b) Modifique la rutina anterior para que reciba como parámetro la dirección inicial del arreglo en R0.
17. ★ Implemente la siguiente rutina a partir de su documentación

```

;-----posDeMaximoEnArreglo
; REQUIERE En R5 la dirección inicial de un arreglo
;           En R7 el tamaño del arreglo.
;           Los valores ocupan 1 celda cada uno.
; MODIFICA ??
; RETORNA en R6 el la dirección del máximo
;-----

```

18. ★ Implemente la siguiente rutina a partir de su documentación

```

;-----proyectarPosicionesPares
; REQUIERE En R5 la dirección inicial de un arreglo
;           Que el arreglo finalice con el valor FFFF
;           Los valores ocupan 1 celda cada uno.
;           En R4 la dirección inicial del nuevo arreglo
; MODIFICA ??
; RETORNA Un nuevo arreglo formado por los valores
;           de las posiciones pares del arreglo original.
;-----

```

19. (a) ★ Al finalizar la fabricación de cada par de zapatos, la máquina que lo produce almacena en una celda de memoria el valor 1 para indicar que el par se fabricó correctamente libre de errores o 0 para el caso contrario.

El siguiente mapa de memoria es un ejemplo de un día determinado de producción:

	...
2000	0001
2001	0000
2002	0001
2003	0001
	...

Escribir un programa que utilice la subrutina `sumaSiEsUno` para contar la cantidad de pares de zapatos que se fabricaron correctamente, analizando los resultados que se volcaron en las celdas 2000 a 2003. El total debe quedar en R2.

- (b) ★ Generalizar el programa anterior para un arreglo de cualquier longitud y cualquier dirección inicial, escribiendo una subrutina que reciba estos datos como parámetros. ¡**Documentela!**
20. (a) ★ En un centro de cómputos se realiza diariamente un chequeo de virus de las computadoras de la sala. Para llevar un registro de la actividad diaria se almacena en una celda de memoria el valor 0 para indicar que la PC se encuentra limpia y en caso contrario el número de virus encontrado. Escriba la siguiente rutina

```

;-----contarCompusLimpias
; REQUIERE En R5 la dirección inicial de un arreglo
;           Que el arreglo finalice con el valor FFFF
;           Los valores ocupan 1 celda cada uno.
; MODIFICA ??
; RETORNA en R7 la cantidad de PC sin virus
;-----

```

(b) Ejecute el siguiente programa

```
MOV R5, 0x1000
CALL contarcompusLimpias
```

Sabiendo que la rutina contarcompusLimpias esta ensamblada en 0xBB00 y teniendo en cuenta el siguiente es el estado de la memoria.

	...
1000	0000
1001	000A
1002	0003
1003	0000
1004	FFFF
	...

Completar la siguiente tabla de cantidad de accesos a memoria:

Instrucción	B.Inst.	B.Op.	Alm.Op.

(c) Escriba la siguiente rutina, tomando como ejemplo la rutina contarCompusLimpias

```
;-----contarCompusVirusBOBO
; REQUIERE En R5 la dirección inicial de un arreglo
; Que el arreglo finalice con el valor FFFF
; Los valores ocupan 1 celda cada uno.
; MODIFICA ??
; RETORNA en R7 la cantidad de PC con el virus BOBO
;-----
```

21. Escriba el programa juventud y la rutina edadEnAños a partir de la información que se provee.

```
;-----edadEnAños
;REQUIERE 1) La fecha de nacimiento (dia,mes,año)
; en los registros R5, R6 y R7 resp.
; 2) La fecha actual en los registros
; R0, R1 y R2
;MODIFICA ?
;RETORNA en R7 la edad de la persona (en años)

;-----juventud
;REQUIERE 1) La fecha de nacimiento de 3 personas,
; almacenada cada una en 3 celdas
; consecutivas a partir de la celda 7890
; 2) La fecha actual en los registros
; R0, R1 y R2
;MODIFICA ?
;RETORNA Un 1 en R7 si una o más de las 3
; personas es mayor a 21
; años y menor a 28, 6 0 en caso contrario
```

22. (a) ★ Implemente la siguiente rutina a partir de su documentación

```
;----- comparador
; REQUIERE Dos valores x e y en bss(16)
; almacenados en los
; registro R6 y R7
; MODIFICA ??
; RETORNA Si x<y suma 1 al registro R0
; Si x=y suma 1 al registro R1
; Si x>y suma 1 al registro R2
;-----
```

(b) ★ Escriba un programa que cuente cuántos valores son menores y cuántos mayores a su predecesor en las celdas 0x7788 a 0x778B. Es decir, se deben hacer las comparaciones:

- la celda 0x7788 (x) con la celda 0x7789 (y)
- la celda 0x7789 (x) con la celda 0x778A (y)
- la celda 0x778A (x) con la celda 0x778B (y)

Los resultados deben volcarse en las celdas 0x6000 (menores) y 0x6001 (mayores).

(c) ★ Generalice el programa anterior para los valores de un arreglo que comienza en la celda 0x0345 y finaliza con el primer valor 0xFFFF.

23. (a) ★ Implemente la siguiente rutina

```
;----- fibonacci
; REQUIERE Un valor n en bss(16) almacenado
; en el registro R7
; MODIFICA ??
; RETORNA El n-esimo valor de la serie de
; Fibonacci en el registro R6
;-----
```

Cada número en la serie de Fibonacci se define como la suma de los dos números anteriores, es decir:

0, 1, 1, 2, 3, 5, 8, 13...

Entonces, si se ejecuta el siguiente programa

```
MOV R7, 0x0005
CALL fibonacci
```

se obtiene el valor 3 en el registro R6.

(b) Escriba un programa que cargue las celdas 0x7788 a 0x778B con los primeros 4 números de la serie de Fibonacci

24. (a) ★ En una fábrica de ventanas las características de los productos de cada venta se codifican mediante cadenas de 16 bits. Cada bit representa una cualidad y se coloca un 1 si cumple con dicha característica, por ejemplo:

- Con el **bit 3** se indica si está pintada.
- Con el **bit 9** se indica si lleva el vidrio de seguridad.

(Nota: Tener en cuenta que los demás bits también representan distintas características, pero para el ejercicio solo nos interesan esos dos).

En caso que la ventana esté pintada o lleve vidrio de seguridad es necesario usar un embalaje distinto, y para esto se pide escribir una rutina que determine colocando un 1 en R7 si el pedido que está en la celda 0x0019 requiere un embalaje *premium*, es decir, si se trata de una ventana pintada o con vidrio de seguridad.

(b) ★ Se tiene un arreglo de pedidos a partir de la celda 0x0001, cuya longitud está en la celda 0x0000. Escriba un programa que recorra

el arreglo para contar en R6 los pedidos que **no necesitan embalaje premium segun el ejercicio anterior**.

25. (a) En un arreglo se tienen codificados los pedidos de una rotisería en cadenas de 16 bits, donde el byte mas significativo representa el tipo de producto y el byte menos significativo la cantidad de unidades. Los códigos de producto son:

- Si bit 15 (mas significativo) es 1: empanadas de carne
- Si bit 14 es 1: empanadas de pollo
- Si bit 13 es 1: empanadas de jamón y queso
- Si bit 12 es 1: pizza napolitana

Por ejemplo:

Cadena	Interpretación
1000000 00000110	6 empanadas de carne
0010000 00001010	10 empanadas de jamón y queso

Se necesita calcular el trabajo de los cocineros, totalizando la cantidad de empanadas de cada tipo. Escriba las siguientes subrutinas:

contarEmpCarne Cuenta en R4 la cantidad pedida de empanadas de carne

contarEmpPollo Cuenta en R5 la cantidad pedida de empanadas de pollo

contarEmpJyQ Cuenta en R6 la cantidad pedida de empanadas de jamón y queso

contarPizzaNapo Cuenta en R5 la cantidad pedida de pizzas napolitanas

Considere que el arreglo de pedidos comienza en la celda 0x5310, y termina con el primer valor 0x0000.

- (b) Considerando la rotisería descrita en el ejercicio anterior, escriba un programa que calcule la ganancia a partir del arreglo de pedidos y un arreglo de precios unitarios que ocupa el rango de celdas 0x5200..0x5203 (pues son 4 productos).

Controlar los programas

Para probar las rutinas/programas que escribiste, escribiremos programas de *test* o prueba como se explicó en la práctica de máscaras

26. ★ Escribí un programa que haga un control de calidad sobre la rutina `maximoValorEnArregloABAB`
27. ★ Escribir un programa que haga un control de calidad sobre la rutina `cantidadNrosPares`.
28. ★ Escribir un programa que haga un control de calidad sobre la rutina `cantidadNrosQuintoBit1`.
29. Escribí un programa que haga un control de calidad sobre la rutina `posDeMaximoEnArreglo`
30. ★ Escribí un programa que haga un control de calidad sobre la rutina `factorial`