

Nº Orden	Apellido y nombre	L.U.	Cantidad de hojas

## Organización del Computador 2

### Primer parcial – 07-10-2014

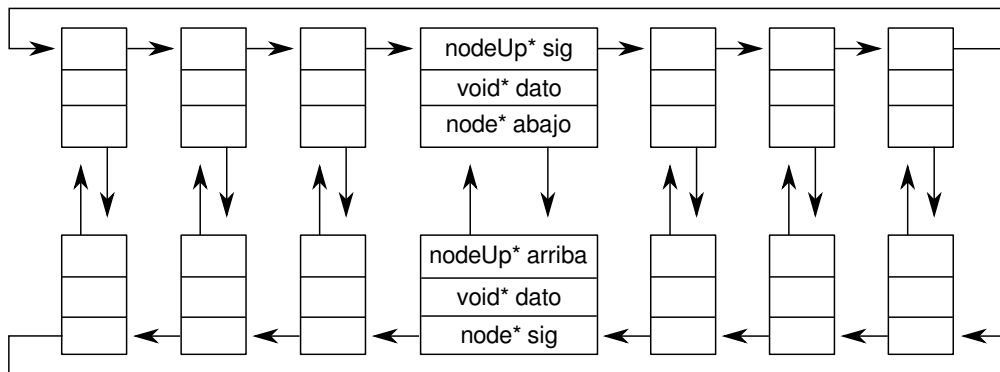
1 (40)	2 (40)	3 (20)	
--------	--------	--------	--

#### Normas generales

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Entregue esta hoja junto al examen, la misma **no** se incluye en la cantidad total de hojas entregadas.
- Está permitido tener los manuales y los apuntes con las listas de instrucciones en el examen. Está prohibido compartir manuales o apuntes entre alumnos durante el examen.
- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con nombre, apellido y LU.
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Los parciales tienen tres notas: I (Insuficiente): 0 a 59 pts, A- (Aprobado condicional): 60 a 64 pts y A (Aprobado): 65 a 100 pts. No se puede aprobar con A- ambos parciales. Los recuperatorios tienen dos notas: I: 0 a 64 pts y A: 65 a 100 pts.

#### Ej. 1. (40 puntos)

Considerar las siguientes dos estructuras de lista `struct node { nodeUP* arriba, void* dato, node* sig}` y `struct nodeUP { nodeUP* sig, void* dato, node* abajo}`. Ambas estructuras se utilizan para formar dos listas doblemente enlazadas como muestra la figura.



- (20p) (a) Implementar en *ASM* la función `borrar_nodo(nodo* ele)`. La misma se encarga de borrar el nodo apuntado y su superior, es decir el `nodoUp` apuntado.
- (20p) (b) Implementar en *ASM* la función `aplicar(nodo* ele, void (*fun)(void*))`. La misma debe aplicar a todos los elementos de las dos listas circulares la función `fun` pasada por parámetro.

Nota: La lista tiene más de 7 elementos.

**Ej. 2. (40 puntos)**

Se poseen dos matrices de datos denominadas  $x$  y  $y$ . Cada matriz tiene exactamente  $512 \times 512$  elementos de un byte sin signo cada uno. Se desea realizar la siguiente operatoria sobre cada una de las matrices.

$$x[i, j] = \begin{cases} 0,32 \cdot x[i, j] + 0,68 \cdot y[i, j] & \text{si } x[i, j] + y[i, j] \text{ es par} \\ 255 & \text{si } x[i, j] + y[i, j] \text{ es impar} \end{cases}$$

$$y[i, j] = \begin{cases} 0 & \text{si } j \text{ es par} \\ y[i, j] & \text{si } j \text{ es impar} \end{cases}$$

- (20p) (a) Escribir el fragmento de código en *ASM* que resuelve la operatoria para  $x[i, j]$ . Escribir el código de solo una interacción del ciclo y explicar el estado inicial de los registros. Este código será citado en el próximo ejercicio.
- (20p) (b) Implementar en *ASM* la función `hacer_cuentas(uint8_t* x, uint8_t* y)`. Esta toma como parámetros las dos matrices y las modifica según corresponda.

Nota: Este ejercicio debe ser resuelto utilizando instrucciones de procesamiento vectorial que calculen la mayor cantidad de datos posibles al mismo tiempo. El índice  $i$  corresponde a filas y  $j$  corresponde a columnas. Todos los calculos se realizan con la versión original de la matriz y luego se actualizan.

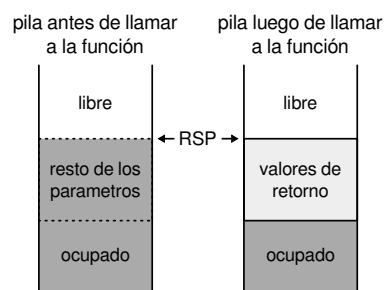
**Ej. 3. (20 puntos)**

Una nueva implementación de la convención para el pasaje de parámetros permite que una función pueda retornar varios valores. La aridad de las funciones se modificará de forma de agregar un conjunto de valores de retorno, es decir:

**type<sub>1</sub>, ..., type<sub>n</sub> functionName ( type<sub>1</sub> name<sub>1</sub>, ..., type<sub>m</sub> name<sub>m</sub> )**

La convención considera tres modificaciones:

- De haber parámetros cargados en la pila, estos serán borrados o suplantados por los valores de retorno
- Los valores de retorno, se escribirán en la pila tal que el valor  $n$  se encuentre en la posición mas significativa de memoria, y el primero en la menos significativa
- Será responsabilidad de la función llamadora liberar la memoria en la pila ocupada por los valores de retorno



- (10p) (a) Implementar la función `int,int sumaryrestar(int a, int b)` que retorna como primer valor la suma entre  $a$  y  $b$ , y luego la resta entre ambos como segundo valor de retorno.
- (10p) (b) Escribir el código necesario para llamar a `sumaryrestar` pasando como parametros los numeros 6 y 2. El resultado debe ser impreso en pantalla utilizando la función `printf`.

Nota: No se permite utilizar memoria dinámica. Tener en cuenta que no necesariamente el resto de los parametros de la función tiene la misma cantidad de posiciones en la pila que los valores de retorno.