

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

Organización del Computador 2

Primer parcial – 29/09/16

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 un árbol binario de búsqueda con la siguiente estructura.

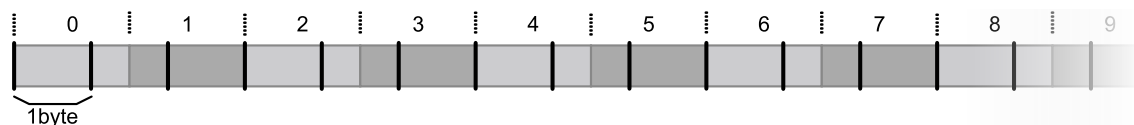
```
typedef struct node_t {
    int value;
    node *derecha;
    node *izquierda;
} node;
```

1. (25p) Construir una función en ASM que determine si el árbol esta ordenado. La misma debe respetar la aridad `void ordenado(node* arbol, node** desordenado)`, tomando como parámetro el puntero al primer nodo del árbol y un doble puntero a nodo donde será almacenado el puntero al subárbol más pequeño que resulte estar desordenado. En el caso que el árbol este ordenado, se escribirá en `desordenado` un cero.
2. (10p) Construir una función en ASM que dado un valor, obtenga el puntero al mismo valor en el árbol. De estar repetido, debe retornar el más cercano a la raíz del árbol. La aridad de la función será `void obtener(node* arbol, int valor, int** valorBuscado)`.
3. (5p) Modificar la función anterior para que retorne el puntero al nodo que contiene el valor. ¿fue necesario realizar alguna modificación?

Nota: Tener en cuenta que para que un árbol binario este ordenado todos los valores del subárbol izquierdo tienen que ser menores al valor de la raíz y todos los elementos del subárbol derecho deben ser mayores o iguales.

Ej. 2. (40 puntos)

Sea un vector de enteros sin signo de 12 bits ordenados de forma contigua como muestra la figura:



Notar que cada valor del vector ocupa un byte y la mitad del siguiente, o medio byte y el siguiente completo.

1. (20p) Construir una función en ASM usando SIMD que dado un arreglo determine si este esta ordenado de menor a mayor.
2. (20p) Construir una función en ASM usando SIMD que dado un arreglo y un número de 12 bits determine si el número pertenece al arreglo.

Nota: Considerar para la solución que el arreglo tiene exactamente 24 elementos.

Ej. 3. (20 puntos)

Se desea modificar la convención C por una nueva denominada convención Z. Esta implementa las siguientes modificaciones sobre la convención C:

- Una función se identifica por un nombre y un conjunto de parámetros escritos con la sintaxis `<nombre>(<param>, ...)`. No es parte de la aridad el tipo del valor de retorno.
- El valor de retorno para toda función es un `uint64` que identifica la cantidad de bytes que se dejan en la pila a la función llamadora.
- Todos los parámetros de una función se pasan por la pila. Se pushean en orden inverso a como figuran en la aridad de la función.

Por ejemplo si se llama a una función `f` que toma un solo parámetro de 4bytes y retorna valores con un total de 24 bytes, la pila se verá de la siguiente forma:



1. (15p) Escribir el código ASM de la función `sumar(int size, int a,...)` respetando la convención Z. Esta función toma una cantidad variable de parámetros enteros donde el primero indica la cantidad de parámetros que recibe la función. La función retorna una cantidad variable de enteros (`int`), siguiendo la siguiente fórmula:

$$r_i = \sum_0^i a_i$$

2. (5p) Escribir el código ASM para llamar a la función anterior con los parámetros 4, 12, 44, 76.

Nota: Considerar que fuera de las modificaciones nombradas, las funciones deben respetar la convención C.