

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

## Organización del Computador 2

### Recuperatorio del Primer Parcial – 19/11/2013

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)

La Navidad esta cerca y como todos los años, el Grinch se dispone a organizar un recorrido de la ciudad para robar regalos. El año pasado su plan se vio truncado cuando un perro le mordió su rabo, lo cual lo llevó a pasar todo el 25 con su cola vendada y recibiendo vacunas antirrábicas, por lo que este año nos pidió que lo ayudáramos con su tarea.

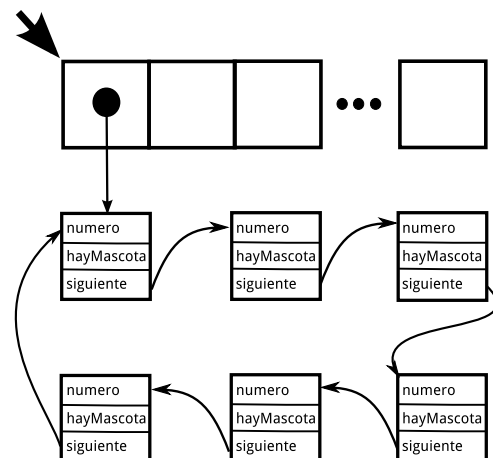
La estructura que representa una GuiaT es una arreglo de punteros a manzanas. Una manzana no es más que un grupo de casas que se apuntan circularmente.

La estructura de una casa es:

```
typedef struct casa_t {
    short numero;
    boolean hayMascotaGuardiana;
    struct casa* siguiente;
} __attribute__((__packed__)) casa;
```

Donde:

```
typedef enum boolean_e { no=0, si=1 } boolean;
```



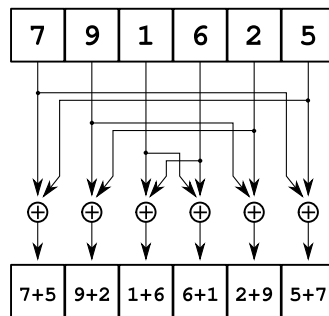
Se pide escribir una función que dada una GuiaT elimine de las manzanas aquellas casas que tienen una mascota guardiana. El prototipo de la función es: `void filtrarGuia(casa** guiat, short n);`

- a) (10p) Escribir en C la función `filtrarGuia`.
- b) (30p) Implementar en ASM de 64 bits la función `filtrarGuia`.

SUGERENCIA: implementar una función auxiliar `casa* filtrarManzana(casa *primera)`

## Ej. 2. (40 puntos)

Se tiene en `rax` el puntero a un vector de 80 elementos de tipo `unsigned char`. Se desea aplicar sobre este vector una operación que sume todos los elementos entre si de forma que los ultimos sean sumados con los primeros y así sucesivamente, para por ultimo ser almacenados en ambas posiciones. A continuación se muestra un ejemplo con un vector de 6 elementos.



La operación se define como:  $P[i] = P[i] + P[size(P) - i - 1]$  para todo  $i$  entre 0 y  $size(P)$ .

- (13p) Implementar la rutina pedida considerando la suma saturada.
- (13p) Modificar la implementación anterior considerando la siguiente función a aplicar,

$$P[i] = \begin{cases} P[i] + P[size(P) - i - 1] & \text{si } P[i] \neq 0 \\ P[i] & \text{si } P[i] = 0 \end{cases}$$

- (14p) Modificar la implementación anterior considerando que el resultado se debe presentar en 16bits y almacenarse en vector aputando por `rbx`.

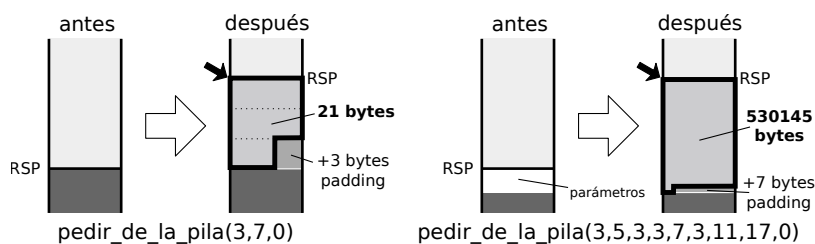
## Ej. 3. (20 puntos)

El equipo de desarrolladores de un O.S. se quedo sin calculadoras de mano, por esta razón decidieron implementar una función que les permita reservar espacio en la pila y liberarlo de forma simple mediante dos funciones.

Ambas reciben una cantidad variable enteros como argumento, y sus aridades son:

`void* pedir_de_la_pila(int a, ... )` y `void retornar_a_pila(int a, ... )`

Para ambas funciones los argumentos se pasan uno a uno, hasta que el último sea cero. La función debe asignar/liberar una cantidad de memoria en la pila igual a la multiplicación de todos estos valores (exceptuando el último). Además, `pedir_de_la_pila` retorna el puntero a la memoria asignada, que debe estar alineada a 8 bytes.



Ejemplos:

`pedir_de_la_pila(3,7,0)`

`pedir_de_la_pila(3,5,3,3,7,3,11,17,0)`

- (15p) Implementar la función `pedir_de_la_pila`.
- (5p) Escribir el código necesario para llamar a la función `pedir_de_la_pila(3,5,3,3,7,3,11,17,0)`

**NOTA:** por simplicidad se considera que todos los parámetros de la función son pasados por pila