

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

Organización del Computador 2

Recuperatorio del Primer Parcial – 01/12/2011

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)

Escriba una función en lenguaje ensamblador que dada una imagen le aplique la siguiente fórmula:

$$\begin{aligned} \max(i, j) = \max(& \text{src}(i-1, j-1) \quad , \quad \text{src}(i-1, j) \quad , \quad \text{src}(i-1, j+1) \\ & \text{src}(i, j-1) \quad , \quad \text{src}(i, j) \quad , \quad \text{src}(i, j+1) \\ & \text{src}(i+1, j-1) \quad , \quad \text{src}(i+1, j) \quad , \quad \text{src}(i+1, j+1)) \end{aligned}$$

$$\text{dst}(i, j) = \text{saturar} \left(\text{src}(i, j) \cdot \left(\frac{255}{\max(i, j)+1} \right) \right)$$

donde *saturar*(*x*) retorna 0 en caso de que *x* < 0 y 255 en caso de que *x* > 255

El prototipo de la función es:

```
void normalizarLocal(unsigned char *src, unsigned char *dst, int alto, int ancho)
```

Aclaraciones:

- La función se debe implementar utilizando instrucciones SSE y se **deben** procesar por lo menos 2 elementos simultáneamente.
- No se debe perder precisión en los cálculos (salvo por las conversiones de punto flotante a entero).
- Fuera del rango en el cual se puede aplicar la fórmula, no se debe modificar la matriz *dst*.
- Puede asumir que *ancho* es múltiplo de un valor conveniente para no tener que manejar casos bordes. **Debe** indicar el múltiplo elegido.
- Las imágenes no tienen padding.
- En cada instrucción SSE se **debe** mostrar el contenido del registro destino.

Ej. 2. (40 puntos)

Una lista circular es una estructura tipo lista, donde se cumple que el último elemento apunta al primero. Sean las siguientes estructuras para almacenar nodos de una lista circular.

```
typedef struct t_nodo_punto {
    unsigned int x;
    unsigned int y;
    nodo_punto *siguiente;
} __attribute__((packed)) nodo_punto;
```

```
typedef struct t_nodo_funcion {
    void* f;
    nodo_funcion *siguiente;
} __attribute__((__packed__)) nodo_funcion;
```

Se desea implementar la función aplicar:

```
void aplicar(nodo_punto* p, nodo_funcion* f)
```

Esta función toma dos punteros a listas circulares, uno de puntos y otro de funciones. La misma se encarga de aplicar las funciones de la lista de funciones a los puntos en la lista de puntos uno a uno. Ambas listas pueden tener distintos tamaños, se deben aplicar las funciones en orden hasta completar todos los puntos. Todas las funciones almacenadas en la lista de funciones tienen la siguiente aridad, `void f(int* x, int* y)`, donde `x` e `y` son dos punteros a los números a modificar.

Ej. 3. (20 puntos)

Sea el siguiente código de la función A.

```
*1  A:    push ebp
      mov ebp, esp
      mov eax, 0
*2      call B
      C:    mov eax, 35
      B:    cmp eax, 35
      jz es35
      mov ebp, [ebp]
*3      call C
*4  es35: add esp, 12
      ret
```

Suponiendo que se llama a la mencionada función, responda:

1. (5 puntos) ¿Que registros son modificados y con que valores terminan? Justificar.
2. (5 puntos) ¿Cuántas veces se ejecuta la instrucción `cmp` y cuántas la instrucción `call`? Justificar.
3. (10 puntos) Indicar el estado de la pila en cada una de las instrucciones marcadas con `*`, luego de ser ejecutadas. Dibuje la misma con todo su contenido.