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

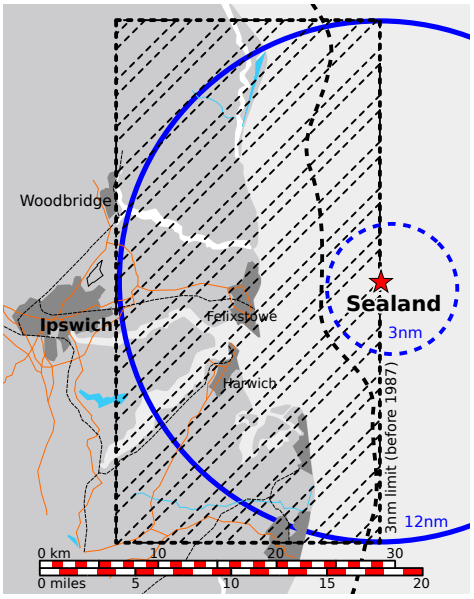
Organización del Computador 2
Primer parcial – 01-10-2013

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.

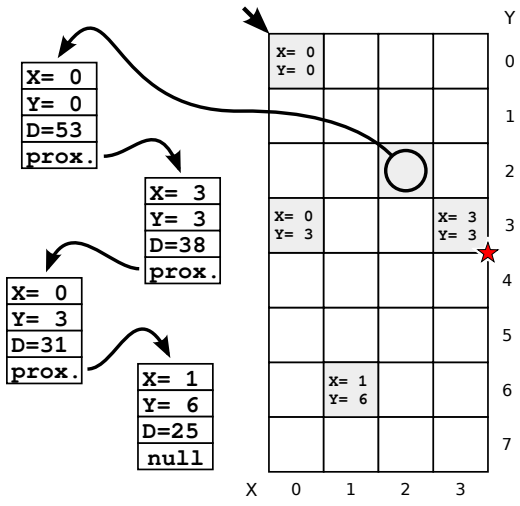
Ej. 1. (40 puntos)



En marzo de 1971 el Principado de Sealand desarrolla un conjunto de tablas con información balística para atacar a Inglaterra desde su fortaleza armada. Estas tablas indican, para cada posición en un plano, que otras posiciones pueden ser atacadas desde ese punto. La estructura utilizada es una matriz de punteros a lista, cada lista contiene el conjunto de posiciones que pueden ser atacadas y el nivel de daño. El tamaño de la matriz es de exactamente 12 nm × 6 nm, espacio sobre el cual se reclama soberanía. En Sealand la milla nautica se divide en 19 partes según sus reglas navales, por lo que la matriz medirá 228 × 114 datos.

La estructura de cada nodo es:

```
typedef struct t_node {
     int pos_x,
     int pos_y,
     unsigned int damage,
     struct node_t *proximo
} __attribute__((__packed__)) node_t;
```



Las coordenadas `x` e `y` que identifican cada una de las areas del mapa, se cuentan desde la esquina superior izquierda del mapa.

a) (20p) Escribir una función que obtenga la lista de las posiciones desde donde se puede atacar a una posición dada. La aridad de la función es: `node* atacameDesde(node **matriz, int x, int y);` (se deben crear nuevos nodos para el resultado)

b) (20p) Escribir una función que filtre la matriz de información balística de forma que solo queden los nodos donde el daño (damage) es mayor que 173. La aridad de la función es: `void dondeRomperMucho(node_t **matriz);`

Ej. 2. (40 puntos)

Se tienen dos arreglos `A` y `B` de 512 valores de tipo `unsigned char` cada uno. Se desea realizar la siguiente operación:

$$A_i = \begin{cases} \texttt{abs}(A_i - B_i) & \text{si } i \text{ es par} \\ (A_i + B_i) - 256 & \text{si } i \text{ es impar y } A_i + B_i > 255 \\ (A_i + B_i) & \text{en otro caso} \end{cases}$$

a) (40p) A continuación se presentan tres alternativas, resolver el ejercicio con dos de ellas indicando cuales alternativas fueron seleccionadas.

I) Utilizando instrucciones de *pack* y *unpack* para filtrar los valores.

II) Utilizando máscaras estáticas o creadas por comparaciones para filtrar los valores y sin usar instrucciones de *shuffle*.

III) Utilizando instrucciones de *shuffle* para acomodar los valores.

Nota: La lógica para recorrer el arreglo puede ser escrita una sola vez

Ej. 3. (20 puntos)

Dada la siguiente estructura:

```
typedef struct t_nodo {
    struct t_nodo *sig,
    char letra
} __attribute__((__packed__)) nodo;
```

Se construye una lista circular de n caracteres. Se desea realizar una función que imprima todos los caracteres de la lista de forma contigua y devuelva la cantidad total de elementos. La aridad de la función es: `int printlista(nodo *lista)`.

a) (10p) Escribir en ASM el código de la función pedida considerando que solo se puede llamar a `printf` una sola vez. Se sabe que la cantidad de elementos de la lista es menor que 100.

b) (10p) Considerando su implementación, si la cantidad de elementos de la lista supera los 100 caracteres ¿Qué problemas puede ocasionar? ¿Cual es la cantidad mínima de elementos que debe tener la lista para generar algún problema?

Nota: No se permite utilizar memoria dinámica.