

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

## TERCERA EVALUACIÓN II TÉRMINO 2009-2010

**Nombre:** \_\_\_\_\_ **Paralelo:** \_\_\_\_\_

### **TEMA 1 (20 puntos)**

Escriba la función **int frecuencia (int numero, int digito)**; la cual retorna el número de veces que **digito** se encuentra en **numero**.

*Por ejemplo:*

numero: -1218 y digito : 1, retorna 2  
numero: 678 y digito : 9, retorna 0

### **TEMA 2 (35 puntos)**

Escribir el procedimiento **generalInformacion** el cual lee el archivo apuntado por **in**, escribe el archivo apuntado por **out** y actualiza el parámetro **result**.

**void generalInformacion (FILE \*in, FILE \*out, char prefijo[4], char result[30]);**

El archivo apuntado por **in** contiene varios registros formados por nombre, apellido y nota de examen separados por punto y coma. El archivo contiene un registro por línea.

Usted deberá escribir en el archivo apuntado por **out** todos aquellos registros cuyo apellido comience con la cadena almacenada en **prefijo**.

Además, se debe actualizar **result** para que almacene el apellido y el nombre del estudiante cuya nota de examen es la más alta entre los datos procesados. Si más de un estudiante tiene la misma nota (la más alta) en **result** sólo se almacenará el último encontrado.

**Formato del archivo apuntado por in:** nombre;apellido;nota

**Formato del archivo de salida apuntado por out:** apellido;nombre;nota

**Formato de cadena result:** apellido nombre

### TEMA 3 (45 puntos)

Una compañía desea implementar la simulación de un juego para hundir barcos en el mar el cual utiliza estructuras de datos.

Usted dispone de la siguiente estructura:

```
typedef struct{
    bool hundido;
    char tipo[30];
}BarcoT;
```

El juego consta de:

- Un repositorio de municiones de tamaño 15, con casillas que contienen balas o que están vacías. Las balas están representadas por el 1 y las casillas vacías por el 0.
- El mar está representado por un arreglo de **BarcoT** de dos dimensiones (8x8). Cada casilla puede tener un barco o estar vacía. Cuando la casilla está vacía significa que hay un barco cuyo tipo es "Vacio".
- El jugador comienza con 100 puntos.

El juego se rige de acuerdo a las siguientes reglas:

1. Llenar el mar con 20 barcos en posiciones aleatorias. Al generar el barco, aleatoriamente se le asignará el campo tipo que puede tener los valores: "Yate", "Crucero", "Pesquero". El resto de celdas del mar tendrán un barco cuyo tipo es "Vacio".
2. Se colocan 11 balas en el repositorio de municiones ubicadas en posiciones aleatorias.
3. El usuario escoge 5 casillas del repositorio de municiones que utilizará para hundir los barcos y el programa debe indicar cuántas de ellas contienen balas.
4. El jugador realiza los disparos en el mar presionando una tecla. En cada disparo se genera una posición aleatoria para el ataque.  
Se debe mostrar un mensaje en pantalla indicando si hundió o no el barco. Si hundió el barco se debe actualizar el campo de la estructura que indica que el barco ha sido hundido. Por cada disparo **fallido pierde** 10 puntos.
5. Al final de la simulación, se debe mostrar el mar usando una X para barcos hundidos, y para los otros barcos la primera letra de su tipo. Por ejemplo: Y para "Yate", V para "Vacio".

**Ud. deberá implementar:**

**a)** El procedimiento **llenarMuniciones (int M [ ], int tam, int nBalas)**, el cual llena el arreglo **M** con el número de balas enviado como parámetro.

**b)** El procedimiento **llenarMar (BarcoT mar [ MAX ][ MAX ], int nBarcos)**, el cual llena el arreglo **mar** con el número de barcos enviado como parámetro.

**c)** El procedimiento **mostrarMar(BarcoT mar [ MAX ][ MAX ])** el cual muestra la matriz **mar** según lo indicado en la regla 5 del juego.

**d)** El programa principal que simule el juego para hundir barcos.

**NOTA:** Use los procedimientos implementados en los literales anteriores.