

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**  
**FUNDAMENTOS DE PROGRAMACIÓN - TERCERA EVALUACIÓN**  
**II TÉRMINO 2012-2013**

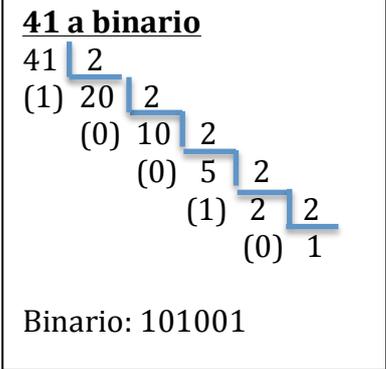
**Nombre:** \_\_\_\_\_ **Matrícula:** \_\_\_\_\_

**TEMA 1 (20 puntos)**

Escriba una función **recursiva** que imprima en pantalla el resultado de convertir un número decimal en binario:

void **dAb**(int num);

```
dAb (41); // imprimirá 101001
dAb (24); // imprimirá 11000
dAb (15); // imprimirá 1111
```



**TEMA 2 (45 puntos)**

Se desea implementar un sistema para la administración de una montaña rusa. Los asientos están representados por una matriz de M filas y N columnas. La posición de cada asiento se definirá con una letra para la fila y un número para la columna. En la matriz los asientos libres están representados por la 'L' y los ocupados por la 'O'.

1	2	3	4	5	6	7	8	9	10	..
O	O	L	L	L	L	L	O	L	L	A
O	L	L	O	L	L	L	O	L	L	B
O	O	L	L	L	L	L	L	L	L	C
										..

**A Ud. Se le ha solicitado implementar:**

- a) El procedimiento **visualizar** (char mr[M][N]), que permita visualizar como en el gráfico qué sitios están disponibles para que una persona pueda decidir dónde sentarse.
- b) La función int **reserva**(char mr[M][N], char fila, int columna), la cual deberá comprobar que la posición del asiento que recibe está libre, devolver 1 y actualizar la matriz, caso contrario devolverá -1.
- c) Un programa principal que muestre el siguiente menú:
  - 1. Ingresar al sistema
    - 1. Reservar asiento
    - 2. Visualizar asientos
    - 3. Liberar todos los asientos
    - 4. Regresar
  - 2. Salir

**Nota:** Reservar asiento le permite al usuario separar con tiempo un asiento en particular. El programa deberá retroalimentar al usuario con mensajes sobre si su asiento fue reservado con éxito o si éste ya se encuentra ocupado. Liberar todos los asientos permite colocar todos los asientos de la matriz en 'L'.

Use los procedimientos y funciones implementados en los literales anteriores.

## TEMA 2 (35 puntos)

Julio César desea mantener oculto el contenido de los mensajes que intercambia con sus socios. Para esto, transforma el mensaje original en un mensaje cifrado utilizando el siguiente esquema: cada letra es interpretada por su posición numérica en el alfabeto, desde la letra A en la posición 1 hasta la Z en la posición 26. Mayúsculas y minúsculas son tratadas de la misma manera y cualquier símbolo que no sea una letra es ignorado. Para transmitir un mensaje, se lo convierte a mayúsculas y cada letra del mensaje se desplaza a través de las letras del alfabeto un número de posiciones previamente establecido para cada uno de sus socios. Por ejemplo, en el caso de un desplazamiento de 5 tendríamos:

**Mensaje original:**    A    B    E    C    E    D    A    R    I    O  
**Mensaje cifrado:**    F    G    J    H    J    I    F    W    N    T

Se obtiene el mismo mensaje cifrado con la palabra “abecedario” o “Abelceda\*riO”.

Para descifrar el mensaje, el procedimiento es el mismo, excepto que el desplazamiento es en el otro sentido. Considere que la información de los mensajes para enviar, los mensajes cifrados recibidos y la especificación con los desplazamientos para cada uno de sus socios se encuentran en los siguientes formatos:

<i>mensajesparaenviar.dat</i>
SOCIO,MENSAJE

<i>desplazamientos.dat</i>
SOCIO,NOMBRE,DESPLAZAMIENTO

<i>mensajescifradosrecibidos.dat</i>
SOCIO,MENSAJE

A continuación un ejemplo:

<i>mensajesparaenviar.dat</i>
1,DESPLIEGUE LAS TROPAS
2,HOLA COMO ESTAS
1,AL ATARDECER
1,MENSAJE URGENTE
2,ESTOY EN LA CIUDAD
3,BIENVENIDO
...
...

<i>desplazamientos.dat</i>
1,ANDRES,5
2,ROMMY,10
3,FERNANDO,8
4,MANUEL,2
5,PEDRO,11
6,JUAN,7
...
...

<i>mensajescifradosrecibidos.dat</i>
2,RYVK
4,EJCS
...
...

**Usted debe implementar:**

1. La función **char \*cifrarMensaje(FILE \*f, int socio, char \*mensaje)** que recibe un puntero al archivo “desplazamientos.dat”, el identificador del socio destinatario del mensaje y el mensaje a cifrar. Esta función debe retornar el mensaje cifrado.
2. El procedimiento **void descifrarMensajes(FILE \*recibidos, FILE \*desplazamientos, int socio)** que recibe un puntero al archivo “mensajescifradosrecibidos.dat”, un puntero al archivo “desplazamientos.dat” y el identificador del socio. El procedimiento descifra y muestra en pantalla todos los mensajes recibidos de ese compañero.