

# Estructura de Datos y de la Información I, Junio 2002

## Examen de Prácticas

Apellidos:

Grupo de Prácticas:

Nombre:

Profesor de Prácticas:

1	2	T

**Nota:** Durante este examen pueden usarse copias de las memorias de las prácticas entregadas, así como libros o cualquier otro material impreso.

1. a. Escribir el código de una función de prototipo: `char ** LeeFicheroNombres(char *nombreFichero, int *numeroNombres);` que lea las cadenas de caracteres incluidas en un fichero de texto. Dicho fichero consta de un registro por línea. El primer registro indica el número total de nombres incluidos en el fichero y a continuación, en líneas sucesivas, cada uno de los nombres (es decir, una línea por cada nombre). La función deberá devolver un array de cadena de caracteres para el cual se aloca memoria dentro de la propia función, y rellenará el número de nombres en la variable entera `numeroNombres`. Debe tenerse en cuenta:

- Leer siempre las líneas del archivo como cadenas de caracteres, convirtiendo a número cuando fuese necesario.

- No reservar más memoria que la estrictamente necesaria para almacenar los nombres.

b. Escribir el código de una función de prototipo

`void LiberaNombres(char **arrayNombres, int numeroNombres);`

que libere la memoria alocada para el array de nombres.

No es necesario implementar el control de errores en ninguna de las funciones de los apartados a) y b). A modo de ejemplo de utilización de dichas funciones se incluye el siguiente programa principal (en el cual tampoco se ha tenido en cuenta el control de errores!):

```
#include <stdio.h>
int main(int argc, char *argv)
{
    char **arrayNombres;
    int i,n;

    arrayNombres=LeeFicheroNombres(argv[1],&n);
    for(i=0;i<n;i++) printf("%s\n",arrayNombres[i]);
    LiberaNombres(arrayNombres, n);
    return (0);
}
```

2. Dada la siguiente estructura para representar una lista doblemente enlazada:

```
typedef struct nodo {
    void *info;
    struct nodo * prev, * next;
} NODO;
```

```
typedef NODO *LISTA;
```

y asumiendo que en el campo `info` se almacenan punteros a cadenas de caracteres, escribir el código de una función de prototipo: `int InsertarNodo(LISTA *lista, NODO * nuevoNodo);`

Dicha función deberá insertar un nuevo nodo en una lista, de forma tal que los nodos queden ordenados según la ordenación alfabética de las cadenas. La función debe devolver 0 si todo ha ido bien, o -1 si encuentra en la lista un nodo que apunte a una cadena idéntica a la que se intenta insertar.

Para comparar las cadenas de caracteres utilizad la función de prototipo `int ComparaCadenas(const char *s1, const char *s2)`. Dicha función comparará las dos cadenas de caracteres `s1` y `s2`, devolviendo un entero menor, igual o mayor que cero cuando `s1` es, respectivamente, menor que, igual a (concordante), o mayor que `s2`.