

# Programación y Estructuras de Datos (PED)

## Examen sobre prácticas Febrero 2008

### 1. Condiciones de entrega

- El examen se entrega a través del servidor de prácticas del DLSI <http://pracdlsi.dlsi.ua.es>.
- Se tiene que entregar un fichero llamado **examen1.tgz** que contenga todos los ficheros del cuadernillo1 (con la estructura de directorios especificada en el enunciado de la práctica), junto con los métodos pedidos en el examen.
- El fichero nombres.txt tiene que contener el nombre del único autor del examen.
- El nombre de la función implementada por el alumno debe coincidir con el nombre propuesto en el enunciado.
- El alumno tiene que implementar su propio fichero de prueba (tad.cpp) para comprobar el código implementado (este fichero no se tiene que entregar).
- El alumno puede añadir a la parte privada las variables y métodos que considere necesarios para la implementación.
- Si se entrega algo que no compila supondrá un 0 en el Cuad1 y en el examen.
- El enunciado se tiene que devolver una vez finalizado el examen.

### 2. Ejercicio

Añadir la siguiente función a la parte pública de TListaCalendario:

```
//Funcion1  
int* Funcion1 (const TVectorCalendario &v)
```

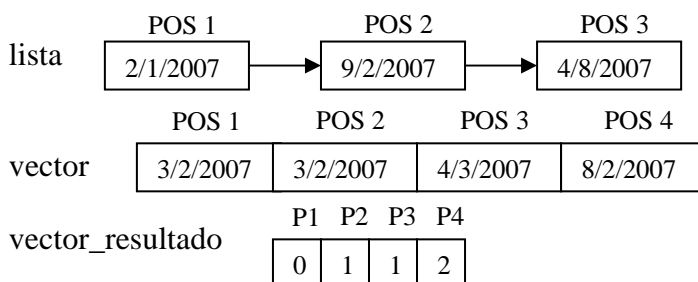
Esta función devuelve un vector de enteros del mismo tamaño que el vector de TCalendario pasado como parámetro.

Para cada posición del vector de TCalendarios comprobamos lo que ocurre con el TCalendario almacenado:

- Para el elemento TCalendario almacenado en la posición i-esima del vector de TCalendarios se comprueba cuántos elementos mayores o iguales que el TCalendario existen entre la posición 1 y la posición i-esima en la lista (ambas posiciones incluidas), el valor obtenido se debe almacenar en la posición i-esima del vector de enteros.

**Nota:** La cabeza de la lista ocupa la posición 1. Si el vector tiene una longitud mayor que la lista, para la posición i-esima del vector que sea mayor que la longitud\_lista se comprobarán todos los elementos de la lista. Si la lista está vacía se pondrá un 0 en todas las posiciones del vector de enteros devuelto. Si el vector es vacío se devuelve un puntero a NULL.

**Ejemplo (está hecho sólo con fechas para simplificar):**



Para vector[p1] miramos:  
lista[p1]  
Para vector[p2] miramos:  
lista[p1], lista[p2]  
Para vector[p3] miramos:  
lista[p1], lista[p2], lista[p3]  
Para vector[p4] miramos:  
lista[p1], lista[p2], lista[p3]