

Apellidos:

Nombre:

Grupo: **A** **B** **C** **D** **E** **F**

Advertencias:

- Duración del examen 2 horas
- No desgrape el cuadernillo del examen.
- Puede utilizar lápiz o bolígrafo indistintamente.
- Las funciones sólo pueden tener un return.
- Sólo puede realizar operaciones de entrada/salida (printf, scanf, gets, etc.) en la función main, salvo que se trate de funciones específicas para leer datos o para mostrar resultados.
- No puede utilizar exit, continue, ni break (salvo en la instrucción switch)
- No puede utilizar variables globales.

Calificación:

1 Vector dinámico	1	
2 Funciones	1	
3 Cadenas	1	
4 Fichero binario	2.5	
5 Vectores	2.5	
6 Matrices	2	
TOTAL	10	

1 Vector dinámico (1 punto)

Suponiendo la siguiente definición de estructura para números complejos, se pide escribir el programa principal (función `main`) que realice las siguientes operaciones:

- Crear un vector dinámico de tipo `T_COMP` para `num` elementos (dato pedido previamente al usuario)
- Inicializar todos los elementos mediante llamadas a `T_COMP Leer(void)`
- Tras finalizar la lectura de todos los elementos, utilizar una función `Mostrar` para que aparezcan por pantalla.
- Liberar la memoria.

Sólo se pide programar la función `main`, no se debe codificar ni `Leer` ni `Mostrar`.

Hay que comprobar los posibles errores de asignación de memoria.

```
/* Nombre...  
  
*/  
typedef struct {  
    double real; //parte real del complejo  
    double imag; //parte imaginaria  
} T_COMP;  
  
T_COMP Leer( void );  
  
int main( void )  
{
```



Fundamentos de Informática 1º Ing. Ind.
Examen: Junio 2008

Junio - 2008

2 Funciones (1 punto)

Escribir las funciones `funcion1()` y `funcion2()` para completar el programa siguiente (**Nota:** pedir al usuario los datos de entrada dentro de cada función) :

...

```
typedef struct
```

```
{
```

```
    int valor;
```

```
    int codigo;
```

```
} T_ARTICULO;
```

```
/* Prototipos */
```

```
int main()
```

```
{
```

```
    T_ARTICULO objeto1;
```

```
    T_ARTICULO objeto2;
```

```
    objeto1=funcion1();
```

```
    printf(" %d %d" , objeto1.valor, objeto1.codigo );
```

```
    funcion2(&objeto2);
```

```
    printf(" %d %d" , objeto2.valor, objeto2.codigo );
```

```
    return 0;
```

```
}
```

```
/* funcion1 */
```

```
/*funcion2 */
```



Fundamentos de Informática 1º Ing. Ind.
Examen: Junio 2008

Junio - 2008

3 Cadenas (1 punto)

Escribir una función con prototipo `int PosicionLetra(char frase[], char letra)` que recibe una frase y una letra y nos devuelva la primera posición de la letra dentro de la frase.

En caso de que la frase no contenga la letra buscada, la función debe devolver -1.

4 Fichero binario (2.5 puntos)

Un archivo binario (productos.dat) ya está creado y contiene toda la información de los productos de una tienda. Los datos de cada producto son: ID (número entero que representa un identificador único del producto, por lo tanto nunca puede estar repetido), Nombre (cadena de caracteres) y el Precio (un número real).

Se quiere disponer de una función que permita modificar el precio de un producto. Escribir una función `CambiarPrecio` que reciba el identificador del producto y su nuevo precio y permita cambiar el precio de este producto. La función `CambiarPrecio` debe abrir el fichero, buscar el producto por código, **y actualizar el registro correspondiente**. Si todo va bien devuelve 0 y si el producto no existe devuelve 1.

NOTA: No se pueden utilizar vectores ni archivos auxiliares porque el objetivo de este ejercicio es utilizar `fseek`.



Fundamentos de Informática 1º Ing. Ind.
Examen: Junio 2008

Junio - 2008

5 Vectores (2.5 puntos)

Un vector de estructuras guarda los datos de los alumnos de una clase. Para cada alumno se usa una estructura T_ALUMNO con estos campos: nombre (cadena de 30 caracteres) y las 10 notas de una asignatura a lo largo de un curso (un vector de tipo real de tamaño 10 para almacenar las notas de los 10 exámenes realizados).

Escribir una función que reciba el vector de estructuras (con datos de todos los alumnos), el número de alumnos y el nombre de un alumno. La función debe obtener la nota media del alumno buscado.

El prototipo de la función es:

```
int NotaMedia(T_ALUMNO *vec, int n, char *nom_alumno, float *media);
```

La función debe devolver uno de los siguientes valores:

0 => no hay error

-1 => el alumno no existe

-2 => el nombre nom_alumno está repetido

Indicar también cómo se realiza la declaración de T_ALUMNO, la declaración de variables en el programa principal así como la llamada a la función NotaMedia. No hay que codificar el programa principal.

```
/* Fi cha
```

```
*/
```

```
/* typedef */
```

```
/*Prototipos */
```

```
int NotaMedia(T_ALUMNO *vec, int n, char *nom_alumno, float *media);
```

```
int main(void)
```

```
{
```

```
    /* Declaración de variables */
```

```
    /* No codificar main */
```

```
    /* Ejemplo de llamada a NotaMedia */
```

```
    return 0;
```

```
}
```

Si gue →

```
int NotaMedia(T_ALUMNO *vec, int n, char *nom_alumno, float *media)
{
```

6 Matrices (2 puntos)

Escribir una función Diagonal () que recibe una matriz cuadrada de enteros (tamaño NxN), el tamaño N y un entero que indica el número de diagonal a calcular. La función debe devolver la media de los valores que están en la diagonal indicada.

La diagonal 0 es la diagonal principal.

La diagonal -1 es la que se sitúa encima de la principal.

La diagonal 1 es la que se sitúa por debajo de la principal.

El identificador de diagonal puede tomar valores desde $-(N-1)$ hasta $(N-1)$. La diagonal $-(N-1)$ está formada por un solo número y la diagonal $(N-1)$ también.

La siguiente figura muestra un ejemplo que consiste en una matriz 5x5.



