

**Ingeniería T. de Informática Sistemas F. y Gestión**  
**Examen de Programación 1. 8 de Julio del 2003**

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_

Grupo \_\_\_\_\_

--	--	--	--	--	--	--	--	--

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• Duración del examen: 3 horas</li><li>• Poner el nombre y apellidos en todas las hojas del examen.</li></ul> | <ul style="list-style-type: none"><li>• Realizar cada problema en páginas diferentes</li><li>• Se valorará la presentación</li><li>• Se valorará la adecuación de las estructuras utilizadas al problema a resolver.</li></ul> |
|---|--|

1. (1 ptos) Escribir las cabeceras de los subprogramas que realizan las siguientes tareas:
  - a. Transformar una fecha del formato ddmmaaaa (string) al formato día (entero), mes (entero) y año (entero).
  - b. Dadas las fechas de nacimiento y actual, en formato ddmmaaaa, calcular la edad (en años) de la persona.
  - c. Dadas una posición inicial y final de un alfil en un tablero de ajedrez, informar sobre la validez del movimiento.
  - d. Dado un elemento de tipo registro de un fichero de alumnos, mostrar por pantalla los datos personales y modificarlos si hace falta.
2. (1 ptos) ¿Qué es un algoritmo de ordenación y para que se utiliza?  
Respuesta:

Explique en unas líneas cómo funciona el algoritmo de **ordenación por inserción**, especificando cuál es la idea que subyace en él.  
Respuesta:

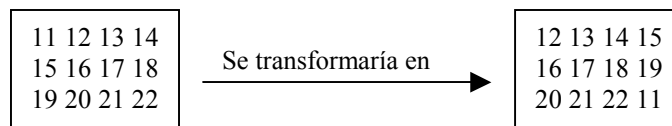
**Ingeniería T. de Informática Sistemas F. y Gestión**  
**Examen de Programación 1. 8 de Julio del 2003**

3. (1 ptos) Se quiere diseñar un programa que ejerza las funciones de una *calculadora* simple sobre números enteros de tamaño indefinido, capaz de realizar las operaciones aritméticas siguientes: *suma*, *resta*, *producto*, *cociente*, *potencia* y *factorial*. Para llevar a cabo la tarea se dispone del tipo de datos adecuado y de dos módulos: *SUMA* (capaz de sumar dos números y devolver el resultado) y *CSIGNO* (que cambia el signo de un número), a partir de los cuales se construye el programa. **Se pide representar el diagrama modular** del sistema resultante, expresando de manera explícita los flujos de datos de las interfaces y las condiciones de entrada bajo las cuales cada módulo puede hacer su tarea (precondiciones).
4. (2 ptos) Escribir una función en Pascal para invertir una cadena de texto. La función recibe una cadena de texto como entrada (“hola y adios”) y devuelve el resultado de invertir los caracteres (“soida y aloh”).
- Hallar una solución iterativa.
  - Hallar una solución recursiva.
- Nota: la función `copy (cadena, posicion, numero)` devuelve la subcadena resultante de extraer de `cadena`, `numero` caracteres a partir de `posicion`.
5. (2 ptos) Dado el tipo siguiente construir un procedimiento `shiftL` que modifique la matriz parámetro, de forma que cada elemento de la misma se desplace a su izquierda, y el primero de cada fila a la última posición de la anterior. El hueco que deja el último elemento de la última fila será ocupado por el primero de la primera.

Type

`tmatriz = array [1..m, 1..n] of integer`

Por ejemplo, la matriz



6. (2 ptos) Los ficheros `vec10.dat` y `vec6.dat` están constituídos por el mismo número de vectores de números enteros de 10 y 6 componentes respectivamente. Se debe construir un fichero `juntos.dat` de vectores de 16 componentes de números enteros, cada uno de los cuales es la fusión de los correspondientes de `vec10.dat` y `vec6.dat`.
- Elaborar el programa Pascal que resuelva el problema, utilizando un procedimiento para la fusión de los vectores.

Por ejemplo:

<pre>vec10.dat 1 2 3 3 5 8 9 5 2 3 3 4 6 7 8 9 1 2 23 3 4 &lt;eof&gt;</pre>	<pre>vec6.dat 1 3 5 9 5 3 3 4 1 2 23 4 &lt;eof&gt;</pre>	<pre>juntos.dat 1 2 3 3 5 8 9 5 2 3 1 3 5 9 5 3 3 4 6 7 8 9 1 2 23 3 4 3 4 1 2 23 4 &lt;eof&gt;</pre>
---	--	---