

## Hoja de ejercicios del Tema 6

1. Utilizando las estructuras de datos definidas en el ejercicio 5 de la hoja de ejercicios del Tema 5 (empleados de una empresa), implementa los siguientes subprogramas:

a) Insertar un nuevo empleado a la plantilla en la posición  $i$  de la lista (con  $0 \leq i \leq \text{contador}$ ).

b) Eliminar el empleado en la posición  $i$  de la lista (con  $0 \leq i < \text{contador}$ ).

2. Un archivo de texto contiene información acerca de los productos que se venden en un almacén. Lo único que se sabe acerca del número de productos es que no puede superar un cierto valor `MaxProductos`. De cada producto se guarda información sobre su código identificador (entero positivo), su precio (real) y el número de unidades existentes (entero positivo). El formato en el que se guarda la información dentro del archivo es el siguiente:

```
id1 precio1 unidades1
id2 precio2 unidades2
...
idN precioN unidadesN
-1
```

a) Declara un tipo `tProducto` que represente la información de un producto y un tipo `tLista` que mantenga la información de todos los productos.

b) Escribe un subprograma que lea los datos del archivo de texto que almacena la información, los guarde en la lista y luego los muestre en la pantalla.

c) Escribe un subprograma que encuentre el producto con máximo valor en el almacén, considerando que el valor del producto  $i$  es  $\text{precio}_i * \text{unidades}_i$ .

d) Escribe un subprograma que dado un identificador de producto a eliminar del almacén, lo busque en la lista y lo elimine actualizando la lista como corresponda.

3. Escribe un programa que lea del teclado una frase y a continuación visualice las palabras de la frase en columna, seguida cada una del número de letras que la componen.

4. Implementa un programa que permita realizar operaciones sobre matrices de  $N \times N$ . El programa debe permitir al usuario la selección de alguna de las siguientes operaciones:
- Sumar 2 matrices.
  - Restar 2 matrices.
  - Multiplicar 2 matrices.
  - Trasponer una matriz.
  - Mostrar una matriz señalando cuáles son los puntos de silla (los *puntos de silla* de una matriz son aquellos elementos de la misma que cumplen ser el mínimo de su fila y el máximo de su columna).

Habrán también dos subprogramas para leer del teclado o mostrar en la pantalla una matriz.

5. Construye las declaraciones de tipos necesarias para almacenar las respuestas (acierto, fallo) a un cuestionario de 20 preguntas. A partir de este tipo de datos define otro que almacene los datos de un examen con las respuestas al cuestionario de 30 estudiantes. Por otro lado, construye una estructura de datos para almacenar el número de veces que cada pregunta ha sido acertada y otra estructura con las calificaciones obtenidas que almacene el número de suspensos, aprobados, notables y sobresalientes (declara un enumerado).

El programa cargará las respuestas de los 30 estudiantes a las 20 preguntas desde un archivo `examen.txt` que contendrá exactamente 30 líneas, cada una con exactamente 20 enteros separados por un espacio. Cada entero será un 1 si la pregunta se ha acertado y un 0 en caso contrario.

Además se implementarán los siguientes subprogramas:

- Dado un cuestionario lo evalúa y devuelve la nota obtenida (entre 0 y 10).
  - A partir de los datos de un examen, rellena una estructura con el número de veces que se ha acertado cada pregunta.
  - A partir de los datos de un examen, rellena la estructura de calificaciones.
6. Una pantalla de texto del PC tiene 25 filas, de 80 columnas cada una de ellas. En cada posición de pantalla se guarda el carácter representado en dicha posición, el color con que se escribe el carácter, el color de fondo donde se visualiza el carácter y si el carácter parpadea o no. Diseña las estructuras de datos necesarias considerando que los colores posibles son: negro, azul, verde, amarillo y rojo.

Construye un subprograma que, dada una fila, devuelva el color de fondo más utilizado en esa fila. Y otro que determine en qué fila y columna se encuentra el primer carácter parpadeante.

7. Como parte de un programa de gestión contable, se necesita una estructura de datos capaz de almacenar los gastos en distintos conceptos producidos cada mes por cada uno de los departamentos. Define tal estructura de datos teniendo en cuenta que los departamentos y los tipos de gastos contemplados son los siguientes:

- ✓ Departamentos: Marketing, Contabilidad, Recursos\_Humanos, Distribución, Ingeniería, Investigación.
- ✓ Conceptos: Salarios, Suministros, Mobiliario, Equipamiento, Otros.

Define subprogramas que, partiendo de una matriz de gastos completa, realicen las siguientes tareas:

- a) Escribir en la pantalla la suma de gastos de cada uno de los departamentos.
- b) Escribir en la pantalla la suma de gastos por cada concepto.
- c) Calcular y devolver el total de gastos de un año.
- d) Construir y devolver un array con los gastos generados cada mes.
- e) Encontrar el valor más alto de la tabla de gastos y devolver el departamento responsable, el tipo de gasto y el mes en que se produjo.

8. En un archivo de texto se encuentran grabados los resultados correspondientes a unas elecciones autonómicas. Cada línea de dicho archivo almacena los datos de una comunidad (máximo 15 comunidades). Éstos son:



- Número de la comunidad autónoma.
- Número de partidos políticos que han obtenido escaños en el parlamento regional correspondiente (máx. 10 partidos).
- Por cada uno de los partidos:
  - ❖ Nombre (una palabra)
  - ❖ Número de escaños obtenidos

Como los nombres de los partidos no pueden tener espacios intermedios nos basta un espacio para separar los distintos campos:

```
1 4 Azul 3 Rojo 4 Verde 1 Negro 1
2 3 Rojo 1 Azul 3 Rosa 1
3 5 Negro 2 Azul 3 Rojo 3 Verde 1 Amarillo 1
4 3 Verde 1 Azul 2 Rojo 1
5 4 Rojo 3 Verde 2 Azul 3 Rosa 1
-1
```

Escribe un programa que muestre los nombres de los partidos vencedores en cada una de las comunidades autónomas, así como el porcentaje de escaños obtenidos. En caso de empate se escribirán los nombres de los partidos empatados.

9. Se quiere desarrollar un subprograma que dadas dos matrices cuadradas de enteros,  $M$  de  $10 \times 10$  y  $P$  de  $3 \times 3$ , compruebe si entre todas las submatrices de  $3 \times 3$  que se pueden formar en la matriz  $M$ , desplazándose por filas o columnas, existe al menos una que coincida con la matriz  $P$ . En ese caso, se indicarán la fila y la columna de la matriz  $M$  en la que empieza la submatriz que coincide. Si hay varias, bastará con indicar la primera.
10. El pirata Pata de Palo desea conseguir el Gran Tesoro sin morir ahogado. La *Isla*, que consiste en un tablero cuadrado de  $N \times N$ , rodeado de agua y con dos puentes en dos esquinas, se puede representar así:

A	G	U	A	
G				A
U				U
A				G
	A	G	U	A

- Escribe un procedimiento `inicializarTablero()` que coloque sobre el tablero al pirata **Pata de Palo** y el **Gran Tesoro** en casillas distintas, determinadas de una forma aleatoria. El **agua** se ha de colocar en los bordes y los **puentes** en las esquinas indicadas.
  - Escribe un subprograma `direccionPirata()` que nos dé la dirección en la que avanzará el pirata (Norte, Sur, Este u Oeste). Un número aleatorio entre 1 y 4 decidirá qué dirección toma el pirata.
  - Escribe un procedimiento `dibujarTablero()` que dibuje el tablero con el formato de la figura.
  - Escribe un programa que permita al pirata **Pata de Palo** intentar buscar el **Gran Tesoro** en menos de 50 movimientos. El pirata recorre los cuadros de uno en uno según la dirección. Al final ha de indicarse si el pirata ha encontrado el tesoro o si se ha ahogado (ha caído al agua).
11. En una panadería se elaboran diariamente los siguientes tipos de productos: colines (0,15), viena (0,65), barra (0,5), baguette (0,9), hogaza de  $\frac{1}{2}$  kg (1,5), hogaza de 1 kg (2,5) y hogaza de 3 kg (4,5). Los fines de semana se elaboran tortas de anises (1,45) y tortas de chicharrones (1,8). Los precios de cada producto se indican entre paréntesis. El panadero desea mantener actualizada la información de una semana sobre la cantidad diaria de productos elaborados, productos vendidos en horario de mañana y productos vendidos en horario de tarde. Considera que el horario de mañana es de 8 a 13 h. y el de tarde es de 16 a 19 h. Realiza las declaraciones de tipos apropiadas para representar la citada información y escribe un subprograma que muestre en la pantalla el dinero obtenido por la venta de todos los productos el último día de la semana.

12. Construye un programa que lea una cadena de caracteres del teclado y construya una nueva cadena con los plurales de las palabras. Suponemos que las cadenas solo contienen nombres y los nombres están separados por un espacio en blanco. Para formar el plural solamente se seguirán las siguientes reglas:

- Los nombres que terminan en vocal añaden s
- Los nombres que terminan en consonante añaden es

13. El juego de las 4 en línea consta de un tablero formado por siete columnas y seis filas. En una partida participan dos jugadores, uno con fichas blancas y otro rojas. Inicialmente todas las posiciones del tablero están libres. Cada jugador coloca alternativamente una ficha en una columna. La ficha colocada cae por su propio peso hasta el fondo de la columna correspondiente (primera fila de la columna libre); por ejemplo, en la figura si el jugador Rojo coloca una ficha en la columna 2, la ficha se coloca en la fila 3. La partida la gana el jugador que coloque en primer lugar cuatro de sus fichas en línea horizontal, vertical o en diagonal. La partida queda en tablas si ninguno de los jugadores es capaz de alinear cuatro fichas después de llenar el tablero.

6	L	L	L	L	L	L	L
5	L	L	L	L	L	L	L
4	L	L	L	■	L	L	L
3	L	L	■	□	■	L	L
2	L	■	□	□	□	■	L
1	L	■	■	□	□	□	■
	1	2	3	4	5	6	7

- a) Codifica las estructuras de datos necesarias para jugar una partida.
- b) Escribe un subprograma `inicializarJuego()` que quite todas las fichas del tablero y prepare el tablero de juego.
- c) Escribe un subprograma `colocarFicha()` que dado un jugador (blanco o rojo) y una columna (1 a 7), coloque la ficha en la posición correspondiente.
- d) Escribe un subprograma `presentarTablero()` que visualice en la pantalla el estado del tablero (como en la figura, excepto la rejilla).
- e) Escribe un subprograma `comprobarGanador()` que, dado un jugador (blanco o rojo) y una determinada casilla (1 a 6, 1 a 7), determine si hay cuatro fichas del mismo jugador alineadas en horizontal a partir de esa posición hacia la derecha.