

T.P. 01 (+ Recursividad – BackTracking - Archivos)

Hacer un programa que permita seleccionar de un menú las siguientes opciones:

- 1) Cargar Matriz.
 - 2) Generar Matriz.
 - 3) Guardar Matriz.
 - 4) Mostrar Soluciones.
 - 5) Salir.
- La opción 1) debe leer desde un archivo secuencial de texto, los datos correspondientes a la cantidad de filas, columnas y la matriz en forma de matriz.
 - La opción 2) debe generar una matriz en N x M entera, y que al azar, la cargue con valores 0 (ceros) y 1 (unos), representando paredes, para los unos y caminos para los ceros.
 - La opción 3) debe almacenar en un archivo secuencial de texto los datos de una matriz (filas, columnas, elementos).
 - La opción 4) deberá encontrar todas las salidas del laberinto, teniendo como condición que siempre la entrada al laberinto se encuentra en la primera fila y la salida del mismo en la última.

0	1	0
1	0	1
0	1	0

En éste ejemplo existen dos entradas al laberinto y dos salidas, y en total hay ocho caminos posibles.

(0,0)(1,1)(2,2) >> (0,0)(1,1)(1,0) >> (0,0)(1,1)(0,2)(1,1)(1,0) >> (0,0)(1,1)(0,2)(1,1)(1,2) ...

El desplazamiento es en cualquiera de los ocho direcciones.

Los datos contenidos en el archivo deben estar dispuestos de la siguiente manera:

```

3
3
0 1 0
1 0 1
0 1 0

```

Donde la primera linea tiene las filas, en la segunda linea las columnas y luego de la segunda los datos de la matriz.

Guía de Trabajos Prácticos de Laboratorio II	2do. Trimestre	2009
--	----------------	------

T.P. 02 (+ Recursividad – BackTracking - Archivos)

Hacer un programa que permita el ingreso de una palabra de 16 caracteres, la distribuya en una matriz de la misma cantidad de filas y columnas de manera que las filas pares se lean de izquierda a derecha, mientras que las impares de derecha a izquierda. Además almacenar cinco palabras de hasta dieciséis caracteres, que mediante un algoritmo recursivo deberá buscar si es posible formar cada una de las palabras con las letra de la matriz, teniendo en cuenta que las letras se encuentren adyacentes y nunca una letra debe utilizarse más de una vez para la formación de la palabra, debiendo mostrar la cantidad de veces que aparece la palabra y las coordenadas que la forman.

Ejemplo: **CONSTITUCIONALES**

C	O	N	S	CONSTITUCIONALES	1
U	T	I	T	TITO	3
C	I	O	N	LEONA	1
S	E	L	A	SITIO	3
				HELP	0

- Donde **CONSTITUCIONALES** se forma una vez.
- **TITO** 3 veces (1,1)(1,2)(1,3)(2,2) // (1,3)(1,2)(1,1)(0,1) // (1,3)(1,2)(1,1)(2,2)
- **LEONA** 1 vez (3,2)(3,1)(2,2)(2,3)(3,3)
- **SITIO** 3 veces (0,3)(1,2)(1,1)(2,1)(2,2) // (3,0)(2,1)(1,1)(1,2)(2,2) // (3,0)(2,1)(1,1)(1,2)(0,1)

Almacenar en un archivo secuencial de texto la matriz en forma de matriz, las palabras a buscar y la cantidad de veces que fueron encontradas cada una de ellas.

También generar un archivo binario para cada palabra a buscar, donde se almacenen letra, posición en x y en y en la matriz. En caso de encontrarse la palabra más de una vez se deberá generar un nuevo archivo.

T.P. 03 (ABML)

Se necesita un sistema que permita manejar los presupuesto de un comercio de venta de componentes para computación.

En la actualidad la empresa se maneja con una lista de proveedores, una lista de los componentes que vende, una lista de clientes y otra de los presupuestos emitidos.

Basado en esos datos se desarrolla el siguiente esquema de manejo de datos:

Productos	Proveedores	Clientes	Presupuestos	Detalle de Presupuesto
<ul style="list-style-type: none"> - Código - Detalle - Precio Venta - Precio Compra - Proveedor 	<ul style="list-style-type: none"> - Código - Nombre - Email - Teléfono 	<ul style="list-style-type: none"> - CUIT - Fecha Alta - Email - Teléfono 	<ul style="list-style-type: none"> - Código - Fecha - Cliente - Total - Descuentos 	<ul style="list-style-type: none"> - Código - Producto - Cantidad - Precio Unitario

El sistema tendrá que permitir la selección de las acciones por medio de un menú:



Se debe tener en cuenta que al momento de dar un alta de presupuesto, se debe dar también altas en el archivo de detalles de presupuestos, que es donde figuran los artículos presupuestados para dicho presupuesto.

Presupuesto:	Cliente:	Fecha:
12345678	Juán Pérez	23/07/09

Cod.	Cant.	Detalle	Precio Unitario	Precio Total
123	1	Mother Asus A8N-VM	130	130
458	1	Procesador AMD 3200 +	110	110
874	2	Memoria DDR 400 Mhz	90	180

Descuentos %	10
Total	378

Nota:

- Los trabajos prácticos deberán ser guardados en una carpeta “**Trim2**” y dentro de ella cada programa llevará el nombre “**labo22gxx-tpnn.c**” (ejemplo: labo22a01-tp01.c).