

# Ejercicios PARCIAL FINAL

## Bloque 1

- Almacene la información concerniente a las ventas de una empresa, de la siguiente forma:  
 Venta: folio(5 dígitos), fecha, nombre del vendedor, total del productos vendidos, identificador del producto vendido (ver producto) , descuento, impuesto y total de la venta.  
 Productos: identificador, cantidad, descripción, precio unitario.  
 El programa debe permitir la captura de las ventas (el numero de productos vendidos, descuento , impuestos y total de la venta son calculados por el programa). Una vez capturada debe mostrar la factura (ver esquema) y una vez confirmada la venta esta debe almacenarse en un archivo.

Folio_____	fecha __/__/__
Vendedor:_____	
Id        cantidad    descripción        precio unitario	total
	Subtotal
	Dscto
	Impuestos
	total

Nota 1: Se aplica 10% de descuento si la venta es de mas 20 artículos.

Nota 2: El impuesto es del 12%.

Nota 3: El programa debe dar alta y baja de productos los cuales se almacenan en un archivo. Considere que en cada venta la cantidad de productos debe actualizarse .

- Escribir un programa que saque por pantalla las 10 palabras que mas se repiten en un archivo (no se cuentan articulos, conectores y/o preposiciones). El nombre del archivo es pedido por pantalla.

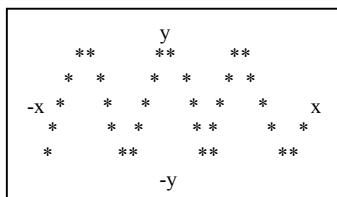
Nota: El nombre del archivo debe proporcionarse mediante la línea de argumentos al llamar al fichero. Es decir:

C:> CtaPalabras fichero.txt

- Considerando que tiene un tipo de dato “struct punto”, donde “x” y “y” son miembros enteros de la misma, y que para formar una línea se requiere de un conjunto de puntos. Escriba un programa que grafique una función lineal, trigonométrica o de potencia, la cuál es elegida por el usuario, donde los valores de “x” y “y” se traducen a coordenadas en pantalla.

Nota1. Utilice la librería conio, donde cada punto graficado es un carácter en pantalla. (ver esquema)

Nota2. La opción (tipo de gráfica) y los valores necesarios para graficar , como los intervalos, deben solicitarse al usuario.



## Bloque 2

- Reparta aleatoriamente a dos jugadores, un mazo de cartas, cada mazo tiene un total de 8 cartas, las cuales contienen (numero, nombre y color de la carta).Mostrar el mazo obtenido por cada jugador en orden ascendente primeramente los rojos, seguidos de los negros.

Numero: As al rey

Nombre: corazones, diamantes, etc.

Color: rojo o negro

Nota1: debe validar los datos según la baraja inglesa

Nota 2: los datos de cada carta están almacenados en un tipo de datos “struct carta”.

5. Dada una estructura gerente(departamento y num. de empleados a cargo), una estructura operativo (gerente, función), almacene la información de cada empleado(nombre, tipo[gerente u operativo], salario). El programa debe permitir el alta y baja de los empleados y debe mostrar el listado de todos los gerentes o bien un listado con el nombre de todos los empleados según lo determine el usuario.  
Nota 1: sabemos que la empresa puede tener un máximo de 50 empleados  
Nota 2: todos los empleados(gerente y operativos) están almacenados sobre una misma variable.
6. Escribir un programa que comprueba que un archivo de código contiene el mismo número de “{“ que de “}” en su código. En caso de que no sea así el programa mostrará un mensaje indicando que el número de “{“ es distinto que el número de “}”.  
Nota1.El programa debe determinar si las llaves fueron cerradas en forma correcta.
7. Crear un programa que lea las siguientes variables proporcionadas desde teclado con el siguiente formato:  
BI6755CC140075---FIESTAFORD, donde,  
Posición de los caracteres    Campo  
1-8                                Matricula  
9                                    Cilindros  
10-12                              Potencia  
12-14                              Año  
15-24                              Modelo  
25-34                              Marca  
Y las introduzca en la correspondiente estructura. El programa deberá mostrar la estructura obtenida para comprobar que la conversión ha sido correcta.  
Nota: Construir una función que muestre por pantalla la estructura recibida. La estructura deberá pasarse por referencia.  
Nota2: Todas las variables convertidas deben ser almacenadas en un archivo. Debe validar que la matricula no debe estar duplicada en caso contrario mostrar un mensaje y no almacenar.
8. Realizar un programa que permita realizar las operaciones básicas (sumar, restar, multiplicar y dividir) números complejos.  
El programa deberá utilizar un tipo “struct complejo” que represente el número complejo en su forma polar, con sus dos componentes módulo y argumento.

### Bloque 3

9. Se sabe que al ingresar un paciente a un hospital los siguientes datos le son solicitados (Nombre, edad, sexo, domicilio, teléfono y numero de seguro), los cuales están almacenados en un tipo struct. Escriba un programa que me permita: listar los nombres de los pacientes , obtener el porcentaje de hombres y mujeres, determinar cuantos pacientes son menores de edad y buscar a un paciente por su numero de seguro social.  
Nota: El programa debe manejar cada opción por medio de una función.
10. Escribir un programa que determine cuantos numeros o letras o signos de puntuación aparecen en un archivo de texto.
11. Dados dos archivos que contienen los datos de dos grupos de alumnos, formar un tercer archivo cuyos elementos resultan de tomar alternativamente un elemento de cada archivo. El nombre del archivo a crear se ha de solicitar por teclado.
12. Escriba una función recursiva para calcular el término n-ésimo de la secuencia de Lucas: 1, 3, 4, 7, 11, 18, 29, 47,...

13. Programe un método recursivo que transforme un número expresado en notación binaria a un número entero.
14. Escriba un programa que resuelva en forma recursiva la función de Acker donde,
- $$\text{Acker}(m,n) = \begin{cases} n+1, & \text{si } m=0 \\ \text{Acker}(m-1),1 & \text{si } n=0 \\ \text{Acker}(m-1,\text{Acker}(m,n-1)) & \text{en otro caso} \end{cases}$$
15. Diseñe e implemente un algoritmo que imprima todas las posibles descomposiciones de un número natural como suma de números menores que él.
- 1 = 1  
2 = 1+1  
3 = 2 + 1  
3 = 1+1+1  
4 = 3+1  
4 = 2+1+1  
4 = 1+1+1+1  
4 = 2+2
- Donde
- $$N = (n-1) + 1$$
- $$N = (n-2) + 2 = (n-2) + 1 + 1$$