

## Ejercicios para proyectos en C++

### Arreglos y Estructuras

1. En una tienda donde sólo hay 10 productos se desea calcular el total de ganancia que produjo dicho negocio este mes. Para ello se cuenta por producto con: Precio del Costo, precio de venta, código y cantidad vendida en ese periodo. Realizar programa que permita calcular las Ganancias de la Tienda.
2. Considere dos matrices A y B, desarrolle un programa que permita realizar la multiplicación de las mismas.
3. Una infantería tiene n cantidad de reclutas. Se desea desarrollar un programa mediante arreglos para determinar cuántos infantes están inscritos en cada curso y con cuantas mujeres y cuantos hombres cuenta la nación en estas especialidades dentro de esta fuerza.
4. Desarrolle un programa que considere una matriz cuadrada de orden  $n \times n$ . El programa debe leer los datos de dicha matriz (números enteros) y encontrar la suma de todos los elementos del borde. Los elementos del borde son aquellos que están en la primera y última fila y columnas.
5. Desarrolle una estructura que guarde los datos de los n alumnos de la materia programación en Lenguaje C, y los muestre por pantalla, la estructura debe tener (cedula, nombre, apellido, edad, profesión, lugar de nacimiento, dirección y teléfono).
6. Escriba un programa que en una estructura guarde los datos de 05 amigos (as) suyos (as) y los muestre por pantalla, utilizando para ello un arreglo de estructura, la misma estructura contendrá (Nombre, Apellido, Edad, Sexo, Dirección, y Teléfono).
7. Desarrolle un arreglo de estructura para un programa que lea una lista de alumnos y las notas correspondientes a una determinada asignatura; el resultado será el tanto por ciento (porcentaje) de los alumnos aprobados y reprobados.
8. Escriba un programa que en una estructura se lleve el control de un inventario de cintas de una tienda de video. Asegúrese de que la estructura incluya el título de la cinta, la longitud de la cinta, el precio de alquiler, la fecha en que se vendió, y la clasificación de la cinta.
9. Desarrolle un programa que lleve el control de los nombres, edades, calificaciones y coeficiente intelectual de 10 estudiantes. Capture los datos desde el teclado utilizando un ciclo for, imprime los datos en la pantalla cuando se termine de capturar los mismos. Emplear Arreglo de Estructuras.

**10.** Desarrolle un programa que en una estructura guarde los datos de los N alumnos de la Asignatura Programación en Lenguaje C; y los imprima, la estructura debe tener (Cedula, nombre, apellido, edad, profesión, lugar de nacimiento, dirección, teléfono).

**11.** Establecer un listado de suscriptores de una revista a través de una estructura, cuyos datos de entrada serán los datos de cada suscriptor (Nombre, Cedula, Mes de Inscripción, Año de Inscripción).

**12.** En una clase, asuma que hay un número de 20 estudiantes. Cada estudiante está registrado en los mismos cursos en un semestre. Al final del semestre, el instructor asigna letras de calificaciones de la A a la E para todos los seis cursos de cada estudiante. Cada uno de los seis cursos es de 3 unidades (horas crédito). El promedio de notas de los estudiantes se calcula en una escala de 10 puntos. Para realizar esto, a continuación se presenta las notas asociadas con cada una de las calificaciones en letras.

**13.** Establezca una estructura para almacenar datos relativos a los N libros o revistas científicas de una biblioteca. Por cada libro o revista científica, mostrar la siguiente información:

- \_ Número de referencia.
- \_ Título
- \_ Nombre del autor
- \_ Editorial
- \_ Clase de publicación (libro o revista)
- \_ Número de edición (sólo libros)
- \_ Año de publicación (sólo libros)
- \_ Nombre de la revista (sólo revistas)

**14.** Desarrolle un programa que inicialice un arreglo de N enteros, y muestre la cantidad y los elementos negativos que se presenta en el mismo.

**15.** Desarrolle un programa que capture tres cadenas, una en mayúscula y las otras en minúscula, luego que la primera cadena la copie al final de la segunda e imprima el resultado, posterior-mente mostrar la longitud de la tercera.

**16.** Se desea calcular el promedio, mínimo y máximo de una lista de notas correspondientes a un curso. Se recibirá el número de alumnos y la nota correspondiente a c/u, y se entregan los resultados en la pantalla

17. Se desea calcular las notas finales de los alumnos de un curso, así como también el promedio, mínimo y máximo de éstas. Se recibirá el número de alumnos y las notas parciales de cada uno de éstos. Las notas parciales corresponden a la I1, I2, T1, T2 y Examen. La nota final se calcula empleando la fórmula:

$$NF = 0,7 * NP + 0,3 * Ex \quad (1)$$

En donde NP es la nota de presentación calculada como:

$$NP = (I1 + I2 + T1 + T2)/4 \quad (2)$$

18. Desarrolle un programa que inicialice un arreglo bidimensional de dimensión 2 x 3, y que imprime y suma sus elementos.

19. La información de todos los empleados de la empresa DATASYSTEM está almacenada en una variable de tipo struct llamada "empleado". La información con que se cuenta de cada empleado es: nombre, sexo y sueldo. Por tanto se pide: Realizar un programa en C++ que lea en un arreglo de estructuras los datos de los N trabajadores de la empresa y que imprima los datos del empleado con mayor y menor salario.

20. A partir de la declaración de las siguientes estructuras realice un programa en C++ que lea el arreglo "ats" y devuelva los datos (nombre, país, deporte) del atleta que ha ganado mayor número de medallas.

21. En una tienda donde solo hay 10 productos se desea calcular el total de ganancia que produjo dicho negocio este mes. Para ello se cuenta por producto con: Precio del Costo, precio de venta, código y cantidad vendida en ese periodo. Realizar programa que permita calcular las Ganancias de la Tienda.

## **Bloque 5: Arreglos o Vectores**

1: Escribe un programa que defina un vector de números y calcule la suma de sus elementos.

2: Escribe un programa que defina un vector de números y calcule la multiplicación acumulada de sus elementos.

3: Escribe un programa que lea de la entrada estándar un vector de números y muestre en la salida estándar los números del vector con sus índices asociados.

4: Escribe un programa que defina un vector de números y muestre en la salida estándar el vector en orden inverso—del último al primer elemento.

5: Desarrolle un programa que lea de la entrada estándar un vector de enteros y determine el mayor elemento del vector.

6: Escribe un programa que defina un vector de números y calcule si existe algún número en el vector cuyo valor equivale a la suma del resto de números del vector.

7: Realiza un programa que defina dos vectores de caracteres y después almacene el contenido de ambos vectores en un nuevo vector, situando en primer lugar los elementos del primer vector seguido por los elementos del segundo vector. Muestre el contenido del nuevo vector en la salida estándar.

8: Hacer un programa que lea 5 números en un arreglo, los copie a otro arreglo, multiplicado por 2 y muestre el segundo arreglo.

### **Bloque 6: Matrices o Tablas**

1: Hacer un programa para rellenar una matriz pidiendo al usuario el número de filas y columnas, Posteriormente mostrar la matriz en pantalla.

2: Realiza un programa que defina una matriz de 3x3 y escriba un ciclo para que muestre la diagonal principal de la matriz.

3: Hacer una matriz de tipo entera de 2 \* 2, llenarla de números y luego copiar todo su contenido hacia otra matriz.

4: Hacer una matriz preguntando al usuario el número de filas y columnas, llenarla de números aleatorios, copiar el contenido a otra matriz y por último mostrarla en pantalla.

5: Realice un programa que lea una matriz de 3x3 y cree su matriz traspuesta. La matriz traspuesta es aquella en la que la columna i era la fila i de la matriz original.

6: Realice un programa que calcule la suma de dos matrices cuadradas de 3x3.

7: Desarrollar un programa que determine si una matriz es simétrica o no. Una matriz es simétrica si es cuadrada y si es igual a su matriz transpuesta.

8: Realice un programa que calcule el producto de dos matrices cuadradas de 3x3.

## Bloque 8: Estructuras

1: Hacer una estructura llamada corredor, en la cual se tendrán los siguientes campos: Nombre, edad, sexo, club, pedir datos al usuario para un corredor, y asignarle una categoría de competición:

- Juvenil  $\leq 18$  años
- Señor  $\leq 40$  años
- Veterano  $> 40$  años

Posteriormente imprimir todos los datos del corredor, incluida su categoría de competición.

2: Hacer una estructura llamada alumno, en la cual se tendrán los siguientes Campos: Nombre, edad, promedio, pedir datos al usuario para 3 alumnos, comprobar cuál de los 3 tiene el mejor promedio y posteriormente imprimir los datos del alumno.

3: Realizar un programa que lea un arreglo de estructuras los datos de N empleados de la empresa y que imprima los datos del empleado con mayor y menor salario.

4: Hacer un arreglo de estructura llamada atleta para N atletas que contenga los siguientes campos: nombre, país, número de medallas. Y devuelva los datos (Nombre, país) del atleta que ha ganado el mayor número de medallas.

5: Hacer 2 estructuras una llamada promedio que tendrá los siguientes campos: nota1, nota2, nota3; y otro llamada alumno que tendrá los siguientes miembros: nombre, sexo, edad; hacer que la estructura promedio este anidada en la estructura alumno, luego pedir todos los datos para un alumno, luego calcular su promedio, y por ultimo imprimir todos sus datos incluidos el promedio.

6: Utilizar las 2 estructuras del problema 5, pero ahora pedir los datos para N alumnos, y calcular cuál de todos tiene el mejor promedio, e imprimir sus datos.

7: Defina una estructura que indique el tiempo empleado por un ciclista en una etapa. La estructura debe tener tres campos: horas, minutos y segundos. Escriba un programa que dado n etapas calcule el tiempo total empleado en correr todas las etapas.

8: Defina una estructura que sirva para representar a una persona. La estructura debe contener dos campos: el nombre de la persona y un valor de tipo lógico que indica si la persona tiene algún tipo de discapacidad. Realice un programa que dado un vector de personas rellene dos nuevos vectores: uno que contenga las personas que no tienen ninguna discapacidad y otro que contenga las personas con discapacidad.

## ESTRUCTURAS O REGISTROS

1.- Una famosa cadena de televisión desea un programa que le permita conocer el tipo de audiencia que tiene en diferentes franjas horarias. Concretamente hay cinco franjas:

- 1.- Antes de las 14 h.
- 2.- De 14 h. a 16,30 h.
- 3.- De 16,30 h a 20 h.
- 4.- De 20 h a 22 h.
- 5.- Después de 22 h.

En cuanto al tipo de audiencia, distinguimos a los televidentes en función de la edad, estableciendo tres rangos:

- 1.- Menos de 20 años.
- 2.- Entre 20 y 50.
- 3.- Más de 50.

El programa irá recibiendo edad y hora (hora:minutos) sobre personas que ven esta cadena y con esa información tendrá que actualizar la estructura de datos AUDIENCIA que será una matriz de 3 filas y 5 columnas. Las filas representan el tipo de audiencia y las columnas representan las distintas franjas horarias.

Se dispone de la estructura T\_hora:

```
struct T_hora{
int h; // de 0 a 23
int m; // de 0 a 59
};
```

2.- Se deben codificar las funciones siguientes:

- Inicializar(A:AUDIENCIA). Inicializa la estructura a 0.
- Incluir(A:AUDIENCIA; edad:entera; hora: T\_hora). A partir de los datos de un televidente (edad y hora a la que ve la televisión), se modifica AUDIENCIA.
- Consulta(A:AUDIENCIA; franja:entera). Devuelve el número de televidentes de una franja a lo largo de todo el día.
- Máximo(A:AUDIENCIA); Devuelve la franja horaria de más audiencia.

Será necesario codificar además dos funciones auxiliares:

- Conv\_hora(h:T\_hora). Recibe como dato una hora y calcula la franja horaria a la que pertenece.
- Conv\_edad(e:entera). Recibe como dato la edad y calcula el rango al que pertenece.

La función principal deberá:

- Inicializar la estructura.
- Modificar AUDIENCIA según los datos de 10 televidentes.
- Consultar la audiencia de una franja.
- Dar la máxima audiencia.
- Se visualizará la estructura cada vez que se crea necesario.

3.- Una empresa se dedica a hacer reformas en pisos y edificios. Para gestionar los proyectos que tienen en marcha han encargado un programa del que se van a implementar algunas funciones y algunas operaciones.  
De cada proyecto se guarda un identificador, el nombre del cliente, el número de días aproximados que dura la reforma, y lo que va a costar. El número máximo de proyectos puede ser de 10.

Escribe la estructura de datos más adecuada a utilizar. Utiliza una estructura, un vector y una variable que indica la ocupación del vector.

- Implementa la función Crear que simplemente pone a cero la variable que indica qué ocupación tiene el vector.
- Implementa la función Insertar que introduce los datos de un nuevo proyecto en la estructura. Debe hacerse de tal forma que los proyectos se puedan recuperar conservando el orden de llegada.
- Implementa la función acabado que, dado el identificador de un proyecto, lo marque como finalizado.
- Implementa la función Total que calcula lo que ha ganado la empresa con los proyectos finalizados.
- Implementa la función Primero que devuelve el identificador del proyecto que lleva más tiempo y que no se ha acabado todavía.
- Implementa la función Ajustado que dado un número de días X, devuelva el identificador del proyecto sin finalizar con una duración prevista menor o igual a X, pero lo más cercana posible. (Es decir si el proyecto 1 tiene una duración de 3, el proyecto 2 de 5 y el proyecto 3 de 8, y nos dan un X = 6, tendrá que devolver el identificador del proyecto 2).
- Implementa la función Main para probar las funciones anteriores.

## Bloque 10: Punteros

1: Comprobar si un número es par o impar, y señalar la posición de memoria donde se está guardando el número. Con punteros.

2: Determinar si un número es primo o no; con puteros y además indicar en qué posición de memoria se guardó el número.

3: Rellenar un array de 10 números, posteriormente utilizando punteros indicar cuales son números pares y su posición en memoria.

4: Rellenar un arreglo con n números, posteriormente utilizando punteros determinar el menor elemento del vector.

5: Pedir al usuario N números, almacenarlos en un arreglo dinámico posteriormente ordenar los números en orden ascendente y mostrarlos en pantalla.  
NOTA: Utilizar cualquier método de ordenamiento.

6: Hacer una función para almacenar N números en un arreglo dinámico, posteriormente en otra función buscar un número en particular.  
NOTA: Puedes utilizar cualquier método de búsqueda (Secuencial o Binaria)

7: Pedir su nombre al usuario y devolver el número de vocales que hay.  
NOTA: Recuerda que debes utilizar punteros.

8: Pedir una cadena de caracteres (string) al usuario, e indicar cuantas veces aparece la vocal a,e,i,o,u; en la cadena de caracteres.  
NOTA: Usar punteros

9: Realice un programa que calcule la suma de dos matrices dinámicas.

10: Realice un programa que lea una matriz dinámica de NxM y cree su matriz traspuesta. La matriz traspuesta es aquella en la que la columna i era la fila i de la matriz original.

11: Hacer una estructura llamada alumno, en la cual se tendrán los siguientes Campos: Nombre, edad, promedio, pedir datos al usuario para 3 alumnos, comprobar cuál de los 3 tiene el mejor promedio y posteriormente imprimir los datos del alumno.  
NOTA: Usar punteros a estructura.

12: Defina una estructura que indique el tiempo empleado por un ciclista en recorrer una etapa. La estructura debe tener tres campos: horas, minutos y segundos. Escriba un programa que dado 3 etapas calcule el tiempo total empleado en correr todas las etapas. NOTA: Usar punteros.

### Paso de Arreglos a funciones

1. Crear un programa que declara un arreglo bidimensional local en la función main() y lo pasa a 3 funciones sucesivas:
  1. Función inputdata que llena el arreglo con datos desde el teclado.
  2. Función sumrows que obtiene y presenta en pantalla las sumas de cada fila del arreglo.
  3. Función sumcolumns que obtiene y presenta en pantalla las sumas de cada columna del arreglo.

### Paso de estructuras a funciones

1. Se ha declarado la estructura de datos DatosPersonales\_t indicada, y en la función main se ha declarado un array de dichas estructuras llamado arrayDPersonales de 100 elementos, así como una variable entera maxElem que indicará en todo momento el número de elementos del array que contienen datos válidos (no tiene por qué ser el máximo de 100).  
Suponga que los datos ya han sido introducidos en el array desde la función main. Implemente una función llamada imprimeDatosPersonales que imprima por pantalla todos los datos personales del array. Indique también cómo se realizará la llamada a dicha función desde la función main.

```
typedef struct
{ char nombre[100]; // nombre y apellidos persona
  char dni[9]; // DNI
```



```

    int edad; // edad
    char sexo; // 'H' para hombre y 'M' para mujer
} DatosPersonales_t;
...
void main()
{
    DatosPersonales_t arrayDPersonales[100];
    int maxElem;
    ...
    // llamada a la función aquí:
    ...
}
// Implementación de la función aquí:

```