

TÉCNICAS DE PROGRAMACIÓN
EXAMEN FINAL
SEMESTRE ACADÉMICO 2024-1

Horario: Todos

Duración: 110 minutos
Elaborado por los profesores del curso.

ADVERTENCIAS:

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación en su mochila.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

INDICACIONES:

- No se pueden usar apuntes de clase ni calculadoras.
- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN Estricto DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 30 líneas de código aproximadamente.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES. NO podrá implementar funciones en el archivo main.cpp, las funciones se deberán implementar en archivos independientes (.h y .cpp).
- No se calificará el código puesto como comentario.
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- **DEBE COLOCAR SU NOMBRE Y CÓDIGO EN EL ARCHIVO main.cpp, DE LO CONTRARIO SE LE DESCOTARÁ 0.5 PUNTOS. NO SE HARÁN EXCEPCIONES.**
- **NO SE PERMITIRÁ LA LECTURA DE DATOS CARÁCTER POR CARÁCTER.**
- **TODA OPERACIÓN DE BÚSQUEDA DEBE REALIZARSE EN FUNCIONES INDEPENDIENTES Y DEBE CONSIDERAR QUE EL DATO BUSCADO NO SE ENCUENTRE.**
- **NO SE PERMITIRÁ EL USO DE ARREGLOS AUXILIARES PARA ALMACENAR LOS DATOS DE LOS ARCHIVOS, SOLO PODRÁ EMPLEAR ARREGLOS EN LOS CAMPOS DE LAS ESTRUCTURAS O DE LOS ARCHIVOS BINARIOS QUE ASÍ SE LE INDIQUE.**

INDICACIONES INICIALES

Siga estrictamente las indicaciones que a continuación se detallan:

- En la unidad de trabajo indicada por los Jefes de práctica (si trabaja en otra unidad, no se calificará el laboratorio y se le asignará como nota cero) Cree allí una carpeta con el nombre "ExamenFinal_2024_1_CO_PA_PN" donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento **se le descontará 2 puntos de la nota final**). **Allí colocará los proyectos solicitados en la prueba**

DEBE LEER TODA LA PRUEBA ANTES DE EMPEZAR A DESARROLLAR EL PROGRAMA

Una cadena de librerías requiere controlar las ventas de sus libros en cada una de sus librerías. Para realizar esta labor cuenta con los siguientes archivos de textos como los que se muestran a continuación:

Libros.csv

ETZ8565,Confesiones de una mascara,Yukio Mishima,138.50
...

El archivo contiene en cada línea los datos de un libro, el código, título, autor y precio del libro.

Clientes.csv

36542155,Perez Rodriguez Julio Alfonso
...

El archivo contiene en cada línea los datos de un cliente, el DNI y el nombre del cliente.

Ventas.txt

78567 28/8/2023 PVZ7181 26290971 65 VDL9379 38724322 72 ...
...

El archivo contiene en cada línea los datos de las ventas de libros: el código de la librería, la fecha de la venta y una lista de: código del libro, DNI del comprador y la calificación que le da el comprador al libro (valor entero entre 0 y 100).

Los archivos no están ordenados.

La cadena librerías requiere contar con una aplicación en C++ que le permita mostrar las ventas por libro en cada librería, en un reporte similar al siguiente:

REPORTE DE VENTAS POR LIBROS Y LIBRERIAS									
LIBRO NO.	1								
CODIGO	TITULO	AUTOR	PRECIO	VENDIDOS	TOTAL.VENTAS	PROM.BUENAS.CALIF	PROM.MALAS.CALIF	PROM.CALIF	
JXC1015	Robinson Crusoe	Daniel Defoe	56.77	25	7898.78	73.5	44.3	65.7	
LIBRERÍA: 56789 TOTAL DE VENTAS: 5565.78									
VENTAS:									
No.	DNI	NOMBRE		CALIFICACION	FECHA				
1	54611233	SANCHEZ ARIAS PEDRO FELIPE IVAN		76	20230320				
2	20238593	WONG LEE FANCISCO JAVIER		36	20230402				
...				
LIBRERÍA: 67665 TOTAL DE VENTAS: 976.7									
VENTAS:									
...				
LIBRO NO.	2								
CODIGO	TITULO	AUTOR	PRECIO	VENDIDOS	TOTAL.VENTAS	PROM.BUENAS.CALIF	PROM.MALAS.CALIF	PROM.CALIF	
...	
LIBRERÍA: ... TOTAL DE VENTAS: ... PROM.BUENAS.CALIF: ... PROM.MALAS.CALIF: ...									
...									
RESUMEN:									
TOTAL DE LIBROS VENDIDOS: 967789									
TOTAL DE VENTAS: 897878789.67									

LOS ARCHIVOS DE TEXTO SE PUEDEN LEER SOLO UNA VEZ.

PREGUNTA 1 (10 puntos): EN ESTA PREGUNTA NO PODRÁ MANEJAR ARCHIVOS BINARIOS.

Cree un proyecto en NetBeans con el nombre: “Estructuras_TP_Pregunta_01” (de no respetar este nombre se le descontarán dos (2) puntos de la pregunta – NO SE HARÁN EXCEPCIONES) y en él desarrolle el programa que resuelva el siguiente problema con las tareas que se describen a continuación. El proyecto debe estar dentro de la carpeta creada previamente.

TODAS LAS CADENAS Y ARREGLOS DEBEN SER DINÁMICOS, Y LAS CADENAS DEBEN ASIGNARSE DE MANERA EXACTA

Las tareas por realizar son las siguientes, las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente:

a) (1.5 puntos) Definir las siguientes estructuras:

- **Cliente** que contenga los siguientes campos: **dni** (valor entero), **nombre** (cadena de caracteres char*), **montoComprado** (valor de punto flotante) y **descuentoFuturo** (valor de punto flotante).
- **Venta** que contenga los siguientes campos: **dniComprador** (valor entero), **nombreComprador** (cadena de caracteres char*), **fecha** (entero en formato aaaammdd) y **calificacion** (valor entero).
- **Libreria** que contenga los siguientes campos: **codigo** (valor entero), **ventas** (arreglo dinámico de tipo Venta), **cantidadVentas** (valor entero), **totalVentas** (valor de punto flotante). Considerar un máximo de 10 ventas por libro.

- **Libro** que contenga los datos del libro y sus ventas en cada librería, sus campos serán: **codigo** (cadena de caracteres char*), **titulo** (cadena de caracteres char*), **autor** (cadena de caracteres), **precio** (valor de punto flotante), **librerias** (arreglo dinámico de tipo Libreria), **cantidadLibrerias** (entero), **totalVentas** (valor de punto flotante), **unidadesVendidas** (valor entero), **cantidadBuenasCalificaciones** (valor entero), **sumaBuenasCalificaciones** (valor de punto flotante), **cantidadMalasCalificaciones** (valor entero), **sumaMalasCalificaciones** (valor de punto flotante). Considerar un máximo de 5 librerías por libro.
- b) (1.5 puntos) Leer los datos del archivo **Clientes.csv** y con ellos llene en un arreglo dinámico de estructuras denominado **arrClientes** de tipo **estructura Cliente**, con la información correspondiente, los campos que no se leen del archivo deben llenarse con valores adecuados. En una función independiente, pruebe la lectura de los datos en un reporte (archivo de texto). El reporte debe tener un título y los datos deben estar bien alineados con encabezados encima de cada columna. El nombre del archivo será “PruebaClientes.txt”, deben mostrarse todos los datos de la estructura.
- c) (1.0 punto) Leer los datos del archivo **Libros.csv** y con ellos llene en un arreglo dinámico de estructuras denominado **arrLibros** de tipo **Libro**, con la información correspondiente, los campos que no se leen del archivo deben llenarse con valores adecuados. Considerar que existen máximo 5 librerías donde se vende el libro y que el libro tiene una buena calificación si tiene un valor mayor a 50, en caso contrario, la calificación es mala.
- d) (1.5 puntos) Emitir el reporte solicitado, con los datos que se tienen, tener en cuenta que los valores que no se hayan completado deben tener los valores adecuados. El nombre del archivo será “PruebaLibros.txt”.
- e) (4.0 puntos) Leer los datos del archivo **Ventas.txt** y con ellos completar la información del arreglo **arrLibros**, tener en cuenta que el arreglo librerías para cada libro, no puede tener información repetida, es decir el código de la librería no se puede repetir. En simultáneo, debe actualizar el arreglo **arrClientes** con la información requerida. Al finalizar la actualización de **arrLibros**, actualice el campo **descuentoFuturo** del cliente en **arrClientes**. Si el cliente ha gastado más de 5000, el descuento será del 30%, si ha gastado más de 3000 y hasta 5000, el descuento será del 20% y si el cliente tuvo alguna compra y gastó hasta 3000 el descuento será del 10%.
- f) (0.5 puntos) Emitir nuevamente los reportes previos: el de la sección b) con el nombre “ClientesActualizado.txt” y el de la sección d) con el nombre “LibrosActualizado.txt”.

Las tareas que no se desarrollen en el orden solicitado no se calificarán.

PREGUNTA 2 (10 puntos): EN ESTA PREGUNTA SERÁ OBLIGATORIO EL USO DE ARCHIVOS BINARIOS, NO ESTÁ PERMITIDO EL USO DE ARREGLOS, SALVO POR LOS INDICADOS EN ESTA PREGUNTA.

Cree un proyecto en NetBeans con el nombre: “ArchBin_ TP_Pregunta_02” **(de no respetar este nombre se le descontarán dos (2) puntos de la pregunta – NO SE HARÁN EXCEPCIONES)** y en él desarrolle el programa que resuelva el siguiente problema con las tareas que se describen a continuación. El proyecto debe estar dentro de la carpeta creada previamente.

Las tareas por realizar son las siguientes, las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente:

- g) (1.5 puntos) Definir las siguientes estructuras:
 - **Cliente** que contenga los siguientes campos: **dni** (valor entero), **nombre** (cadena de caracteres), **montoComprado** (valor de punto flotante) y **descuentoFuturo** (valor de punto flotante).
 - **Venta** que contenga los siguientes campos: **dniComprador**(valor entero), **nombreComprador**(cadena de caracteres), **fecha** (entero en formato aaaammdd) y **calificacion** (valor entero).

- **Libreria** que contenga los siguientes campos: **codigo** (valor entero), **ventas** (arreglo de tipo Venta), **cantidadVentas** (valor entero), **totalVentas** (valor de punto flotante). Considerar un máximo de 10 ventas por libro.
 - **Libro** que contenga los datos del libro y sus ventas en cada librería, sus campos serán: **codigo** (cadena de caracteres), **titulo** (cadena de caracteres), **autor** (cadena de caracteres), **precio** (valor de punto flotante), **librerias** (arreglo de tipo Librería), **cantidadLibrerias** (entero), **totalVentas** (valor de punto flotante), **unidadesVendidas** (valor entero), **cantidadBuenasCalificaciones** (valor entero), **sumaBuenasCalificaciones** (valor de punto flotante), **cantidadMalasCalificaciones** (valor entero), **sumaMalasCalificaciones** (valor de punto flotante). Considerar un máximo de 5 librerías por libro.
- h) (1.5 puntos) Leer los datos del archivo **Cientes.csv** y con ellos llenar en un archivo binario denominado **Cientes.bin**, que guarde **estructuras de tipo Cliente**, la información correspondiente, los campos que no se leen del archivo deben llenarse con valores adecuados. En una función independiente, pruebe la lectura de los datos en un reporte (archivo de texto). El reporte debe tener un título y los datos deben estar bien alineados con encabezados encima de cada columna. El nombre del archivo será "PruebaCientes.txt", deben mostrarse todos los datos de la estructura.
- i) (1.0 punto) Leer los datos del archivo **Libros.csv** y con ellos llenar un archivo binario denominado **Libros.bin** que guarde **estructuras de tipo Libro**, la información correspondiente, los campos que no se leen del archivo deben llenarse con valores adecuados. Considerar que existen máximo 5 librerías donde se vende el libro y que el libro tiene una buena calificación si tiene un valor mayor a 50, en caso contrario, la calificación es mala.
- j) (1.5 puntos) Emitir el reporte solicitado, con los datos que se tienen, tener en cuenta que los valores que no se hayan completado deben tener los valores adecuados. El nombre del archivo será "PruebaLibros.txt".
- k) (4.0 puntos) Leer los datos del archivo **Ventas.txt** y con ellos completar la información del archivo **Libros.bin**, tener en cuenta que el arreglo librerias para cada libro, no puede tener información repetida, es decir el código de la librería no se puede repetir. En simultáneo, debe actualizar el archivo **Cientes.bin** con la información requerida. Al finalizar la actualización de **Libros.bin**, actualice el campo **descuentoFuturo** del cliente en **Cientes.bin**. Si el cliente ha gastado más de 5000, el descuento será del 30%, si ha gastado más de 3000 y hasta 5000, el descuento será del 20% y si el cliente tuvo alguna compra y gastó hasta 3000 el descuento será del 10%.
- l) (0.5 puntos) Emitir nuevamente los reportes previos: el de la sección b) con el nombre "ClientesActualizado.txt" y el de la sección d) con el nombre "LibrosActualizado.txt".

Las tareas que no se desarrollen en el orden solicitado no se calificarán.

Al finalizar el examen, comprima la carpeta que contiene los proyectos empleando el programa Zip que viene por defecto en el Windows, no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares y súbalo a la tarea programada en PAIDEIA para este examen.

CRITERIOS DE CALIFICACIÓN:

1. Si el programa entregado presenta más de tres errores de sintaxis serán calificados sobre la mitad del puntaje.
2. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido.
3. Se descontará 15% de la nota si no se colocan comentarios relevantes, incluyendo un encabezado al inicio del programa en el que se indique el nombre del autor, la fecha, y una descripción de lo que hace programa).
4. No se calificará el código puesto como comentario.
5. No se calificarán aquellas funciones implementadas en el archivo main.cpp

San Miguel, lunes 01 de julio del 2024