

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. Examen

(Primer Semestre 2024)

Indicaciones Generales:

- Duración: 170 minutos (2 horas con 50 minutos).

NO SE PERMITE EL USO DE APUNTES DE CLASE, FOTOCOPIAS NI MATERIAL IMPRESO

- No se pueden emplear **variables globales**, **NI OBJETOS** (con excepción de los elementos de `iostream`, `omanip` y `fstream`). **NO PUEDE UTILIZAR LA CLASE `string`**. Tampoco se podrán emplear las funciones de C que gestionen memoria como `malloc`, `realloc`, `memset`, `strdup`, `strtok` o similares, igualmente no se puede emplear cualquier función contenida en las bibliotecas `stdio.h`, `cstdio` o similares y que puedan estar también definidas en otras bibliotecas. **NO PODRÁ EMPLEAR PLANTILLAS EN ESTE LABORATORIO**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ERICTO DISEÑO DESCENDENTE. **Cada función NO debe sobrepasar las 20 líneas de código aproximadamente**. El archivo `main.cpp` solo podrá contener la función `main` de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. En el archivo `main.cpp` deberá colocar un comentario en el que coloque claramente su nombre y código, **de no hacerlo se le descontará 0.5 puntos en la nota final**.
- El código comentado NO SE CALIFICARÁ. De igual manera NO SE CALIFICARÁ el código de una función si esta función no es llamada en ninguna parte del proyecto o su llamado está comentado.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.

SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO.

NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES DADAS EN LA PRUEBA

- **Puntaje total:** 20 puntos.

Puntaje total: 20 puntos

INDICACIONES INICIALES

Cree un proyecto de C++ en NetBeans siguiendo estrictamente las indicaciones que a continuación se detallan:

- La unidad de trabajo será **t:** (Si lo coloca en otra unidad, no se calificará su laboratorio y se le asignará como nota cero)
- Cree allí una carpeta con el nombre **"EX01_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 3 puntos de la nota final). **Allí colocará los proyectos solicitados en la prueba.**

Se tienen tres archivos del tipo CSV, los cuales se describen a continuación:

Archivo de libros (.csv)
ETZ8565,Confesiones de una mascara,Yukio Mishima
LQL0880,Contra el fascismo,Umberto Eco
...

Código del libro, nombre, autor

Archivo de clientes (CSV)
36542155,Perez Rodriguez Julio Alfonso
67128011,Quispe Huaman Luisa Hilda
...

DNI y nombre del cliente.

Archivo de ventas (CSV)
PVZ7181,26290971,28/8/2023,65
ICX1503,27912250,5/9/2023,23
AVN3710,54602211,27/8/2023,71
...

Código del libro, DNI del comprador, fecha del venta, puntaje del libro (1-100).

PUNTEROS MÚLTIPLES

PREGUNTA 1 (6 puntos)

Elabore un proyecto denominado "PunterosMultiples1Examen01Pregunta01" y en él desarrollará el programa que dé solución al problema planteado. **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL.**

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include "PunterosMultiples1Examen01Pregunta01.h"

int main(int argc, char** argv) {
    char **ventasLibroCodigo, ***libro;
    int **ventasDniFechaPuntaje;

    cargarVentas (ventasLibroCodigo, ventasDniFechaPuntaje, "Ventas.csv");
    pruebaDeCargaDeVentas (ventasLibroCodigo, ventasDniFechaPuntaje, "PruebaVentas.txt");
    cargaLibros (libro, "Libros.csv");
    pruebaDeCargaDeLibros (libro, "PruebaLibros.txt");

    return 0;
}
```

**NO PUEDE CAMBIAR
ESTE CÓDIGO**

Implemente las funciones **cargarVentas**, **cargarLibros**, **pruebaDeCargaDeVentas** y **pruebaDeCargaDeLibros**, la función "cargarVentas" y "cargarLibros" deben leer los datos de los archivos "Ventas.csv" y "Libros.csv" y colocarlos en las estructuras como se muestra en la figura No. 1, según corresponda. El archivo CSV debe leerse una sola vez, en todo el proyecto. Los espacios de memoria asignados para todos los datos deben ser **dinámicos y por incrementos de 5 en 5** (salvo para las cadenas de caracteres que deben ser exactas). **De emplearse otro método de asignación de memoria NO se asignará puntaje en esta pregunta.** Los libros deben ser ordenados por el título empleando el qsort de stdlib.

Las funciones "pruebaDeCargaDeVentas" y "pruebaDeCargaDeLibros" deben emitir un reporte que pruebe, de manera clara, **bien alineada** y con encabezados adecuados encima de cada columna, la carga correcta de los datos. Los datos de una venta o de un libro deben aparecer en una línea, no se permite que aparezcan como en una ficha.

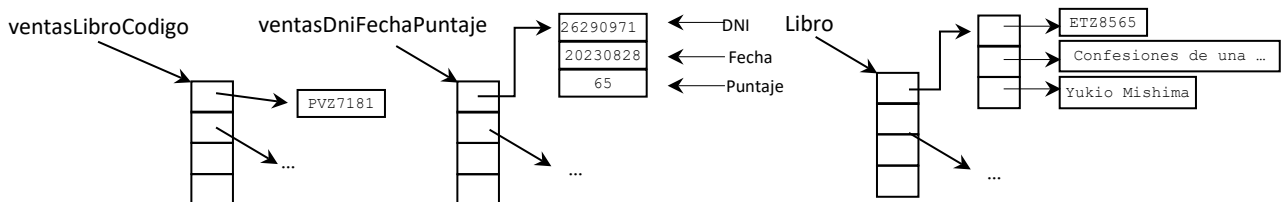


Figura No. 1

NOTA A TENER EN CUENTA:

Se le entregará una biblioteca estática con la solución de esta pregunta, usted tendrá la opción de desarrollar esta pregunta o simplemente emplear la solución dada para solucionar las siguientes preguntas del examen, en este último caso no se le calificará esta pregunta.

PREGUNTA 2 (7 puntos)

PUNTEROS GENÉRICOS

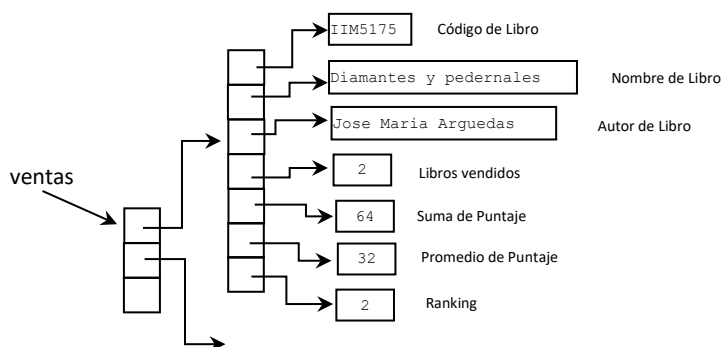
Elabore un proyecto denominado "PunterosGenericos1Examen01Pregunta02" y en él desarrollará el programa que dé solución al problema planteado. **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL.**

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

La codificación en rojo corresponde a la pregunta 1

**NO PUEDE
CAMBIAR ESTE
CÓDIGO**

La función **"muestraventas"** implementada en la biblioteca MuestraVoid debe ser empleada para la impresión de prueba de la estructura.



NOTA A TENER EN CUENTA:

Se le entregará una biblioteca estática con la solución de esta pregunta, usted tendrá la opción de desarrollar esta pregunta o simplemente emplear la solución dada para solucionar las siguientes preguntas del examen, en este último caso no se le calificará esta pregunta.

PUNTEROS A FUNCIÓN

Página 3 de 5

Desarrolle la biblioteca **ColaPrioridadGenerica**, que brinde soporte al **ranking** que aparece en el main, con las funciones necesarias para su soporte, como son las funciones **generacola**, **encola**, **desencola** y **colavacia**. La función **generacola** es una función que se encarga de crear la estructura que representa a la cola. Se recomienda generar tres nodos iniciales que apunten a nullptr con la finalidad que cada prioridad sea representada desde el inicio. Como se ve en la imagen, la cola está formada por 4 direcciones de memoria, la primera apunta a la cabeza de la cola, la segunda a la cola de la prioridad 3, la tercera a la cola de la prioridad 2 y la cuarta a la cola de la prioridad 1.

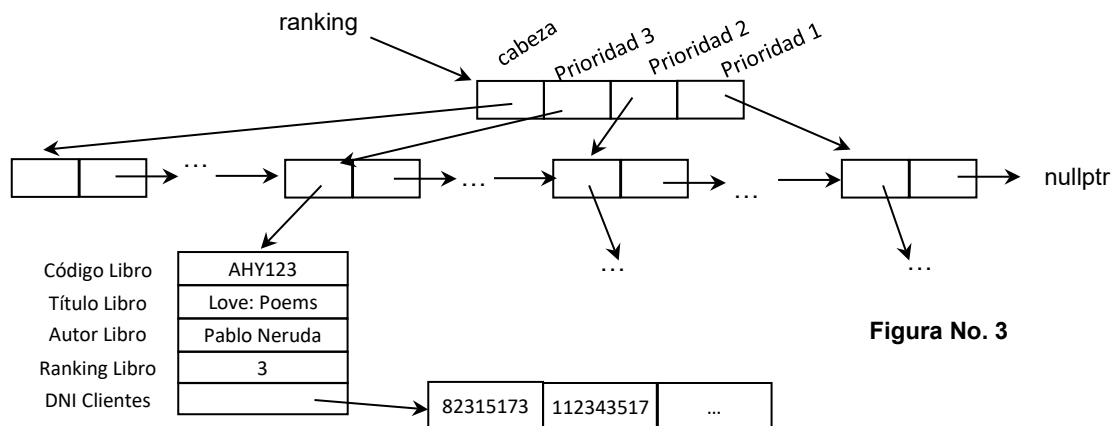


Figura No. 3

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include "PunterosMultiples1Examen01Pregunta01.h"
#include "PunteroVoid.h"
#include "MuestraVoid.h"
#include "PunterosFuncion1Examen01Pregunta02.h"
#include "ColaPrioridadGenerica.h"
#include "Registros.h"
int main(int argc, char** argv) {
    char **ventasLibroCodigo, **libro;
    int **ventasDniFechaPuntaje;
    void *ventas;
    void * ranking;
    cargarVentas(ventasLibroCodigo, ventasDniFechaPuntaje, "Ventas.csv");
    cargarLibros(libro, "Libros.csv");
    cargaventas(libro, ventas);
    actualizaventas(ventasDniFechaPuntaje, ventasLibroCodigo, ventas);
    muestraventas(ventas);
    cargarranking(ranking, crearegistro, ventas);
    muestraranking(ranking, imprimirregistro, "rankings.txt", ventas);

    return 0;
}
```

**NO PUEDE
CAMBIAR ESTE
CÓDIGO**

Implemente en "PunterosFuncion1Examen01Pregunta02.h":

1. **cargarranking** (5ptos): Para este proceso deberá ubicar en la cola los libros por orden de ranking (3, 2, 1). Use los punteros de la cola de prioridades para la inserción. En la cabeza de la cola deberá estar ubicado el libro con mejor ranking (ranking 3).
2. **muestraranking** (2ptos) La función **muestraranking** recibe el nombre del archivo donde realizará la impresión del reporte y las variables extras para obtener un reporte de libros en cada ranking. Cada libro deberá ser impreso en una sola línea. Los headers y el orden de la impresión serán tomados en cuenta para la calificación.

NOTAS IMPORTANTES:

- En ningún caso se permitirá desarrollar dos preguntas en el mismo proyecto. De hacerlo no se calificará la segunda y/o tercera pregunta.

- Las marcas de fin de datos solo podrán ser cero o nulo.
- Toda tarea de búsqueda debe ser desarrollada en una función independiente. Toda función de búsqueda debe prever que el dato buscado no se encuentre en la estructura empleada.

Al finalizar la práctica, comprima la carpeta de su proyecto 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.**

Profesores del curso: Rony Cueva
 Erick Huiza
 Miguel Guanira

Erasmus Gómez
Heider Sánchez

San Miguel, 14 de mayo del 2024.