

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJES DE PROGRAMACIÓN 1**

**6ta práctica (tipo b)**  
**Segundo Semestre 2021**

**Indicaciones Generales:**

- Duración: 110 minutos.

Obligatoriamente los alumnos deberán mantener en todo momento el AUDIO Y VIDEO de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen y la revisión de los trabajos que estén desarrollando. De tener algún problema deberán hacérselo saber de inmediato al profesor de su horario por correo. De no hacerlo, no se aceptarán reclamos alegando que no oyeron las indicaciones.

- No se pueden emplear variables globales, estructuras, ni objetos (con excepción de los elementos de `iostream`, `io manip` y `fstream`). Tampoco se podrán emplear las funciones `malloc`, `realloc`, `strdup` o `strtok`, 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. Tampoco podrá hacer uso de plantillas.
- No se puede emplear variables estáticas.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO 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 cada archivo que implemente en los proyectos (`.h` y `.cpp`) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontará 0.5 puntos por archivo.
- NO SE CALIFICARÁN aquellas funciones desarrolladas en el mismo archivo que la función `main`.
- El código comentado NO SE CALIFICARÁ.
- Los proyectos deben obligatoriamente desarrollarse en NetBeans bajo el sistema operativo Windows. No se revisarán los proyectos desarrollados en otros sistemas operativos o IDEs.
- 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. ESTO Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA `.cpp` O `.h` CON FECHA U HORA DE CREACIÓN ANTERIOR A LA EVALUACIÓN SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

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

- **Puntaje total:** 20 puntos.

**Cuestionario:**

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 5 del curso: "Listas Genéricas". En este laboratorio se trabajará con punteros sin tipo, así como la implementación de TAD's.

Deberá elaborar un proyecto denominado "`LAB06_ColaPrioridad`" y en él desarrollará el programa que dé solución al problema planteado. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.

Se tienen los archivos que se describen a continuación:

Pedidos.csv
99215341E,ZM20053,Raffo Arias Surami Annie
15129661E,QA53897,Wong Martel Madeleine
33958767A,SK15788,Aquino Toro Rocio
...

Carné/Tipo,Código de libro,Nombre

Numeros.txt
20
320
16
...

Numero

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

```
#include <cstdlib>
#include "BibGenerica.h"
#include "bibimpresion.h"
#include "FunConEnteros.h"
#include "FunConRegistro.h"

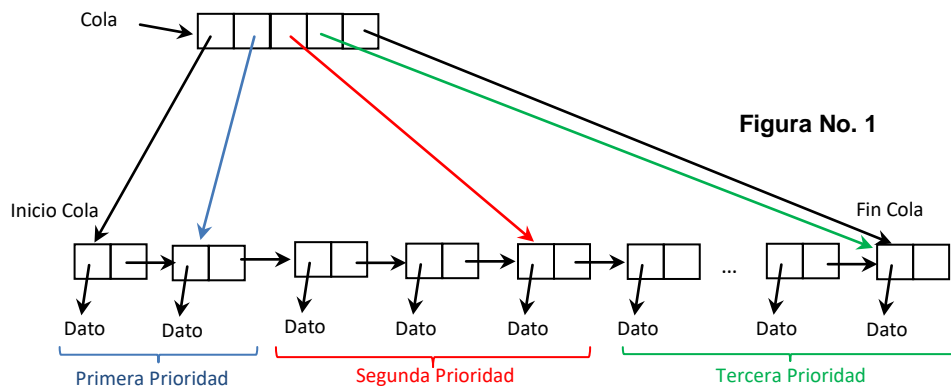
using namespace std;
int main(int argc, char** argv) {
    void *cola;
    creacola(cola, leenumero, prioridadnumero, "numeros.txt");
    imprimirnum(cola, "Reporte.txt");
    descargarcola(cola, muestranumero, "Orden.txt");

    creacola(cola, leepedido, prioridadpedido, "Pedidos.csv");
    imprimirped(cola, "Reppedidos.txt");
    descargarcola(cola, muestrapedido, "Ordenpedidos.txt");

    return 0;
}
```

**NO PUEDE  
CAMBIAR  
ESTE CÓDIGO**

Para solucionar este problema se le solicita que defina una lista genérica que se comporta como una cola. La cola no podrá ser manipulada como una lista simplemente ligada. La misma debe tener la siguiente estructura:



Esta estructura debe contar con una biblioteca **BibGenerica**, con las funciones necesarias para su soporte, como son las funciones **creacola**, **encola**, **desencola**, **colavacia** y **descargarcola**. La función **creacola** recibe el nombre del archivo a ser cargado en la cola, considerando las 3 posibles prioridades que poseen los datos. Para realizar esta operación se debe considerar una función **encola** con una complejidad  $O(1)$ , en otras palabras, **la cola no puede recorrerse para su llenada**. Por tal motivo se debe contar con las 3 direcciones que marcan el fin de cada grupo de acuerdo con su prioridad, como se muestra en la figura 1.

Las funciones **imprimirnum** e **imprimirped** reciben el nombre de los archivos donde realizaran la impresión y se encuentran implementados en la biblioteca estática **bibimpresion** (libbibimpresion.a).

#### **Pregunta 1 (12 puntos)**

Desarrolle la biblioteca **BibGenerica**, que brinde soporte a la cola que aparece en el main de acuerdo con la figura 1. Para probar el buen funcionamiento de esta biblioteca genérica, desarrolle la biblioteca **FunConEnteros** la cual debe trabajar **solo** con los enteros. Los números deben encolarse de acuerdo con el grupo de prioridad al que pertenece, determinado por la función **prioridadnumero** el cual se define de la siguiente forma:

- Si el número es menor a 50, pertenece a la prioridad 3.
- Si el número es mayor o igual a 50 y menor igual a 100, pertenece a la prioridad 2.
- Si el número es mayor a 100, pertenece a la prioridad 1.

Dentro de cada grupo prioritario los elementos van en el orden en que llegan a la cola.

La función **imprimirnum** mostrará el contenido de la cola y el último elemento de cada grupo de prioridad:

320
200
71
58
70
51
20
16
46
19
26
Prioridad 1:200
Prioridad 2:51
Prioridad 3:26

Finalmente debe desarrollar la función **descargarcola** que debe desencolar cada elemento de la cola e imprimirlo en el orden de prioridades, numerando cada elemento, de la siguiente forma:

Orden	Numero
1	320
2	200
3	71
4	58
5	70
6	51
7	20
...	

### **Pregunta 2 (8 puntos)**

Una vez comprobado el buen funcionamiento de su biblioteca genérica, desarrolle la biblioteca **FunConRegistro**, donde deberá cargar todos los datos del archivo Pedidos en cada nodo de la cola, utilizando la función **prioridadpedido** para la clasificación, de acuerdo con las siguientes prioridades:

- Los docentes pertenecen a la prioridad 1.
- Los estudiantes son parte de la prioridad 2.
- Los administrativos son parte de la prioridad 3.

Al igual que la pregunta anterior, luego de cargar la cola puede probarla utilizando la función **imprimirped**, con el siguiente orden en el registro: código del usuario, tipo, libro y nombre del usuario. La función **descargarcola** deberá desencolar cada nodo e imprimir un reporte similar al que se muestra a continuación:

OrdenPedidos				
ORDEN	TIPO	CODIGO	NOMBRE	LIBRO
=====				
1	D	77001234	Diaz Correa Irma Virginia	LM05731
2	D	83265244	Pairazaman Alamo Moises Miguel	PL70268
3	D	83265244	Pairazaman Alamo Moises Miguel	FW74175
4	D	49516360	Delgado Gonzales Guillermo Alfonso	SV17719
5	D	37751890	Shirakawa Miranda Madeleine	FW74175
6	D	81082011	Arroyo Gordillo Maria Heli	FW74175
...				

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**. Luego súbalo a la tarea programa en Paideia para este laboratorio.

Profesores del curso: Miguel Guanira  
Rony Cueva

San Miguel, 5 de noviembre del 2021.