

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

6ta práctica (tipo b)
Primer 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_ListaGenerica`" 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 tres archivos del tipo CSV, los cuales se describen a continuación:

Conductores.csv
30018447,FUENTES QUISPE CINTHIA DELIA
32517791,ZORRILLA LARA ARTURO
...

licencia, nombre del conductor

RegistroDeFaltas1.csv	
35912665,O2Z-308,22/07/2019,156	licencia,placa,fecha,infracción
32244565,M8W-853,18/11/2019,335	
...	

RegistroDeFaltas2.csv	
30018447,O1C-880,25/08/2020,102	licencia,placa,fecha,infracción
78153392,A7R-205,18/12/2020,672	
...	

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

```
#include <cstdlib>
#include "BibliotecaListaGenerica.h"
#include "ListaConEnteros.h"
#include "ListaConRegistro.h"

using namespace std;
int main(int argc, char** argv) {
    void *lista1,*lista2;

    creaLista(lista1,leenum,cmpnum,"RegistroDeFaltas1.csv");
    creaLista(lista2,leenum,cmpnum,"RegistroDeFaltas2.csv");
    uneLista(lista1,lista2);
    imprimeLista(lista1,imprimenum,"Repnum.txt");

    creaLista(lista1,leeregistro,cmpregistro,"RegistroDeFaltas1.csv");
    creaLista(lista2,leeregistro,cmpregistro,"RegistroDeFaltas2.csv");
    uneLista(lista1,lista2);
    imprimeLista(lista1,imprimeregistro,"Repfalta.txt");

    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

Para solucionar este problema se le solicita que defina una lista genérica simplemente ligada. La misma debe tener la siguiente estructura:

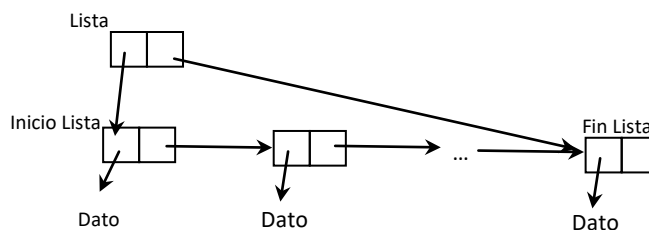


Figura No. 1

Esta estructura debe contar con una biblioteca **BibliotecaListaGenerica**, con las funciones necesarias para su soporte, como son las funciones **creaLista**, **insertaLista**, **listavacia**, **uneLista** e **imprimeLista**. La función **creaLista** recibe el nombre del archivo a ser cargado en la lista. Con el archivo **RegistroDeFaltas1** debe crear la **lista1**, con el archivo **RegistroDeFaltas2** debe crear la **lista2**, con estas 2 listas cargadas debe realizar la unión de estas estructuras utilizando la función **uneLista**. La lista unificada debe almacenarse en la **lista1**, siguiendo la estructura de la figura1, además **la complejidad de esta operación de unión debe ser $O(1)$** . La función **imprimeLista** recibe el nombre del archivo donde realizará la impresión de la lista según lo solicitado en cada pregunta.

Pregunta 1 (10 puntos)

Desarrolle la biblioteca **BibliotecaListaGenerica**, que brinde soporte a las listas que aparecen en el main de acuerdo con la figura1. Para probar el buen funcionamiento de esta biblioteca genérica, desarrolle la

biblioteca **ListaConEnteros** la cual debe trabajar **solo** con los enteros representados por la fecha de los archivos de Registro de Faltas. La representación de la fecha como número debe ser de la siguiente forma: **Por ejemplo, para la fecha 25/08/2020 se debe representar como el número 20200825**. Los números deben insertarse en cada lista de forma ascendente. Finalmente debe imprimir la lista unificada en el archivo brindado como parámetro de la función, considerando que las fechas no se repitan en el reporte de control. De la siguiente forma:

```
20190102
20190103
20190104
...
20200310
20200311
20200312
...
```

Pregunta 2 (10 puntos)

Una vez comprobado el buen funcionamiento de su biblioteca genérica, desarrolle la biblioteca **ListaConRegistro**, donde deberá cargar todos los datos del archivo Registro de Faltas en cada nodo de la lista, ordenados por fecha. Al igual que la pregunta anterior, luego de cargar las dos listas debe unir las en una sola y emitir un reporte similar al que se muestra a continuación:

Reporte			
Año: 2019Mes: 01			
FECHA	LICENCIA	NOMBRE	FALTA
=====	=====	=====	=====
2019/01/02	16477908	BACA CAMACHO MILAGROS NATALIA	115
2019/01/02	47844324	VILCA LEE NESTOR HUGO	329
...			
2019/01/05	81485316	JUSTINO MARTINEZ ENRIQUE	327
...			
...			
...			
=====	=====	=====	=====
Año: 2019 Mes: 04			
FECHA	LICENCIA	NOMBRE	FALTA
=====	=====	=====	=====
2019/04/03	80457845	NORABUENA MOYA HECTOR	113
2019/04/04	89941095	ARCA LARRIEGA LUCERO YLIANA	147
2019/04/05	82031355	EGUSQUIZA BERROCAL DORIS NANCY	122
...			
...			

El reporte muestra la información considerada en ambos archivos, dividido por bloques de Año y Mes. Para obtener el nombre de los conductores puede recorrer el archivo **Conductores.csv** todas las veces que sea necesaria.

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, 11 de junio del 2021.