

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

3ra 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**, ni **objetos** (con excepción de los elementos de `iostream`, `iomanip` 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.
- 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 4 del curso: "Funciones y alcance de variables". En este laboratorio se desarrollará una **biblioteca estática de funciones** en la que se implementen sobrecargas de operadores y funciones que le permitan manejar las infracciones de tránsito registradas por una entidad reguladora.

Cree una **carpeta** denominada "**LP1_Lab03_2021-2_Biblioteca**", dentro de ella cree dos **carpetas** denominadas "**P1_CreaBiblioteca**" y "**P2_UsaBiblioteca**", en estas últimas colocará los proyectos solicitados en las partes 1 y 2 respectivamente de este laboratorio. **DE NO COLOCAR ALGUNO DE ESTOS REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.**

PARTE 01 (14 puntos): CREACIÓN DE LA BIBLIOTECA ESTÁTICA

Se solicita que desarrolle una biblioteca estática denominada "**Biblioteca_2021_2**" en la cual se definirán las estructuras de datos que describen a continuación, además definirá una serie de operadores sobrecargados que permitirá manejar las estructuras **(DEBE EMPLEAR DE MANERA**

OBLIGATORIA LOS NOMBRES DE LAS ESTRUCTURAS Y SUS CAMPOS DADOS DE LO CONTRARIO SE DESCONTARÁ UN TERCIO DEL PUNTAJE). Las estructuras se definen a continuación:

➤ **Para manejar los Usuarios de la Biblioteca:** La estructura (**struct**) se denominará **"St_Usuario"** y deberá contener lo siguiente: 1) un campo denominado **carne**, este campo almacenará el número de carné de un usuario (**int**), 2) un campo denominado **tipo** definido por una cadena de caracteres (almacenará el tipo de usuario: **"ESTUDIANTE"**, **"ADMINISTRATIVO"** o **"DOCENTE"**), 3) un campo denominado **nombre** definido por una cadena de caracteres, 4) un campo denominado **libPrest**, definido por un arreglo de estructuras de tipo **St_LibroPrestado**, considerar un máximo de 20 elementos, 5) un campo denominado **numLibPrest** (**int**) que llevará la cuenta de los libros que tiene el usuario en calidad de préstamo, 6) un campo denominado **condicion** definido por una cadena de caracteres (almacenará el estado del usuario: **"HABILITADO"** o **"INHABILITADO"**).

➤ **Para manejar los libros prestados:** La estructura (**struct**) denominada **"St_LibroPrestado"** deberá contener lo siguiente: 1) un campo denominado **codigoLib**, para almacenar el código del libro definido por una cadena de caracteres, 2) un campo denominado **fechaDeRetiro** (**int**) que almacenará una fecha en que retiró el usuario el libro, en formato AAAAMMDD, 3) un campo denominado **deuda** (**bool**) que almacenará el valor de **true** si el libro superó el plazo de devolución y **false** si el usuario aún tiene plazo para seguir contando con el libro.

➤ **Para manejar los libros de la Biblioteca:** La estructura (**struct**) se denominará **"St_Libro"** deberá contener lo siguiente: 1) un campo denominado **codigo** definido por una cadena de caracteres, 2) un campo denominado **titulo** definido por una cadena de caracteres, 3) un campo denominado **stock** (**int**) con la cantidad de libros que posee la biblioteca, 4) un campo denominado **precio** (**double**), con el valor del libro, 5) un campo denominado **prestados** (**int**) con la cantidad de libros que se han prestado, 6) tres campos definidos por arreglos de estructuras tipo **St_UsuarioPrestamo**, denominados **estudiantes**, **docentes** y **administrativos** que guardarán los datos de los usuarios que tiene en su poder el libro, según el tipo de usuario. 7) tres campos denominados **cantEst**, **cantDoc** y **cantAdm** (**int**) que guardarán la cantidad de usuarios que prestaron el libro según el tipo.

➤ **Para manejar los libros de la Biblioteca:** La estructura (**struct**) se denominará **"St_UsuarioPrestamo"** deberá contener lo siguiente: 1) un campo denominado **carne** (**int**), 2) un campo denominado **tipo** (**char**) que guardará la letra correspondiente al tipo, 3) un campo denominado **fechaDeRetiro** (**int**) que almacenará una fecha en que retiró el usuario el libro, en formato AAAAMMDD, 4) un campo denominado **deuda** (**bool**) que almacenará el valor de **true** si el libro superó el plazo de devolución y **false** si el usuario puede seguir contando con el libro.

Las operaciones que la biblioteca estática permitirá realizar a través de sobrecargas de operadores y funciones se definen a continuación:

➤ **Lectura:**

- Sobrecargando el operador **>>** de modo que permita leer **un** libro de un archivo de textos. La operación (**arch >> libro;**) involucrará un archivo de textos y una variable de tipo **"St_Libro"**. La sobrecarga deberá devolver **true** si se pudo leer la información y **false** si se llegó al final de archivo. Una línea de archivo tendrá la siguiente forma:

```
VG01047  Lenguaje_de_programacion_C++  12  325.75
(codigo - titulo - stock - precio)
```

- Sobrecargando el operador **>>** de modo que permita leer **un** usuario de un archivo de textos. La operación (**arch >> usuario;**) involucrará un archivo de textos y una variable de tipo **"St_Usuario"**. La sobrecarga deberá devolver **true** si se pudo leer la información y **false** si se llegó al final de archivo. Una línea de archivo tendrá la siguiente forma:

```
30018447E  CUEVA_FUENTES_CINTHIA_DELIA
carné:NumeroTipo - nombre)
```

El resto de los campos deberán ser inicializados adecuadamente.

➤ **Agregación:**

- Sobrecargando el operador **+=** de modo que permita agregar **un** préstamo a **un** usuario. La operación (**usuario += libroPrestado;**) colocará los datos de un préstamo (**St_LibroPrestado**) al

final de su lista de préstamos de un usuario, modificando el número de libros prestados (**numLibPrest**). No debe modificar los campos que no están involucrados en la operación.

- Sobrecargando el operador += de modo que permita agregar un préstamo a un libro. La operación (**libro += usuarioPrest**;) colocará los datos de un préstamo (**St_Usuario_Prest**) al final de la lista de usuarios (según el tipo) modificando la cantidad de libros prestados y la cantidad del usuarios correspondiente. No debe modificar los campos que no están involucrados en la operación.
- **Verificación de deudas en usuarios:** sobrecargando el operador <= de modo que permita verificar si el usuario debe uno o más libros. La operación **usuario <= fecha** (donde fecha es un int) deberá marcar con **true** todos aquellos libros que el usuario tiene en su poder más días de los permitidos (ver función **"numeroDeDias"** descrita más adelante), con false aquellos que no, además colocar en el campo condición la palabra correspondiente si se le inhabilitó o no en caso de deuda. Un docente puede tener en su poder un libro solo 15 días, un estudiante 7 días y un administrativo solo 3.
- **Verificación de deudas en libros:** sobrecargando el operador <= de modo que permita verificar si un libro tiene usuarios con deudas. La operación **libro <= fecha** deberá marcar con true todos aquellos usuarios que tienen en su poder el libro más días de los permitidos y con false aquellos que no.
- **Impresión:**
 - Sobrecargando el operador << de modo que permita imprimir la información de un usuario. La operación (**arch << usuario**;) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo **"St_Usuario"**. El formato será el siguiente:

```
30018447
Arca/Amezquita/Edric-Ronald
Estudiante
INHABILITADO
AB12345          03/09/2021
VG765430        28/08/2021      Deuda
...              ...              ...
```

El reporte debe estar perfectamente tabulado (**sin usar el carácter '\t'**).

- Sobrecargando el operador << de modo que permita imprimir la información de un libro. La operación (**arch << libro**;) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo **"St_Libro"**. El formato será el siguiente:

```
Titulo : Lenguaje_de_programacion_C++
Codigo : VG01047      Autor : Raul/Disman
Edicion : 2010        Precio : 325.75
Stock : 12            Prestados: 3
=====
Usuarios que lo tienen en su poder:
=====
DOCENTES:
=====
No.      Carne      Fecha de retiro  Observacion
1)       32546712      4/09/2021
=====
ESTUDIANTES:
=====
ADMINISTRATIVOS:
=====
No.      Carne      Fecha de retiro  Observacion
1)       54312890      23/08/2021      Deuda
2)       65225177      05/09/2021
=====
```

El reporte debe estar perfectamente tabulado (**sin usar el carácter '\t'**).

Consideraciones:

La solución debe contemplar la elaboración de: 1) un proyecto de implementación y prueba de las sobrecargas, denominado **"Biblioteca_Puebas_de_Fuentes"**, 2) un proyecto que genere la biblioteca estática denominada **"Biblioteca_Compilacion"** y 3) un proyecto donde se pruebe la biblioteca ya compilada **"Biblioteca_Puebas_Compilada"**. La prueba de las sobrecargas para el primer y tercer proyecto deben ser hecha lo más simple posible pero que muestre claramente son correctas (**No se debe solucionar aquí el problema de la pregunta 2**). Los tres proyectos deberán colocarse en la carpeta **"P1_CreaBiblioteca"**.

El código función **"numeroDeDias"** se le proporcionara junto con la prueba y debe incorporarlo a la biblioteca. La función devuelve el número de días entre dos fechas (int) con formato AAAAMMDD.

PARTE 2 (6 puntos): REUTILIZACIÓN DE LA BIBLIOTECA ESTÁTICA.

Desarrolle en la carpeta “P2_UsaBiblioteca” un proyecto denominado “Biblioteca_Aplicacion” en el cual se utilizará obligatoriamente las sobrecargas de la biblioteca estática (compilada) “Biblioteca_2021_2”. La pregunta no se evaluará si no se usa la biblioteca compilada (.a). El proyecto ejecutará las tareas descritas a continuación utilizando las sobrecargas definidas en la biblioteca:

- a) Leer los dato de los usuarios contenidos en un archivo de textos como se muestra a continuación y los coloque en un arreglo de estructuras del tipo “St_Usuario”:

30018447E	FUENTES_QUIISPE_CINTHIA_DELIA
32517791D	ZORRILLA_LARA_ARTURO
...	

- b) Leer los dato de los libros de la Biblioteca contenidos en un archivo de textos como se muestra a continuación y los coloque en un arreglo de estructuras del tipo “St_Libro”:

VG01047	Lenguaje_de_programacion_C++	12	325.75
MG75319	La historia del mundo	25	187.55
...			

- c) Leer uno a uno los préstamos que se han realizado y agréguelos a los arreglos creados anteriormente. El archivo que es similar al siguiente:

26873214A	XN48844	22/6/2021	Carné / Código del libro / fecha en que se retiró
11223344D	VN17150	14/3/2021	
...			

- d) Leer una fecha en formato dd/mm/aaaa y determinar las deudas en usuarios y libros.
e) Emitir un reporte en el que se muestren los datos todos los libros.

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, 24 de septiembre del 2021.