PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1
1er. Examen
(Primer Semestre 2023)

Indicaciones Generales:

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

SOLO ESTÁ PERMITIDO EL USO DE APUNTES DE CLASE. NO PUEDE UTILIZAR FOTOCOPIAS NI MATERIAL IMPRESO, TAMPOCO PODRÁ EMPLEAR HOJAS SUELTAS.

- No se pueden emplear **variables globales, estructuras, ni objetos** (con excepción de los elementos de iostream, iomanip y fstream). **No puede utilizar la clase (o el tipo de datos) string**. 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. **NO PODRÁ EMPLEAR PLANTILLAS EN ESTE EXAMEN**
- 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 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 muestres resultados o que estos no sean coherentes en base al 60%.
- **EN NINGUNA PREGUNTA PODRÁ UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES**
- **NO PODRÁ COLOCAR LOS DATOS EN ARREGLOS AUXILIARES A LOS INDICADOS EN LAS PREGUNTAS O LOS PROPIOS DE LOS MÉTODOS SOLICITADOS**
- 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

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 **"CO_PA_PN_EX01_2023_1"** 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 cuatro archivos del tipo CSV, los cuales se describen a continuación:

Cursos.csv

```
INF263,Algoritmia,3.75,35030611,INGA FLORES CESAR ADOLFO
MEC270,Procesos De Manufactura,4,83265244,PAIRAZAMAN ALAMO MOISES MIGUEL
……
```

Código, Nombre,Céditos, Código del profesor, Nombre del profesor

HistoriaDeNotas.csv

```
202123703,MEC270,202302,14
202315643,MEC270,202202,13
…
```

Cód. Alum. Cód. Curso,Ciclo, Nota

Alunos.csv

```
202123703,GAMARRA TABORI PAUL RONAL,S,30,G5
202119153,MENDOZA ARIAS HENRY,G4
202113758,SOMA INGA DIALY,V, G1
…
```

Código, Nombre, Modalidad (Virtual [V], Semi presencial [S] + porcentaje o presencial [vacío], Escala

Escalas.csv

```
G5,666.90
G1,282.30
G3,454.20
G2,362.00
G4,556.70
```

Escala, valor del crédito

PUNTEROS MÚLTIPLES

PREGUNTA 1 (6 puntos)

Elabore un proyecto denominado "Pregunta01Examen01PunterosMultiples" 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 "Preg1_Funciones_de_cursos_y_alumnos.h"

int main(int argc, char** argv) {
   char ***cursos, **alumnos_nom_mod;
   double *cursos_cred, escalas[5];
   int *alumnos_cod, **alumnos;
   cargarCursosYEscalas (cursos, cursos_cred, escalas,"Cursos.csv", "Escalas.csv");
   pruebaDeCargaDeCursos(cursos, cursos_cred, "PruebaCursos.txt");

   cargarAlumnos (alumnos_cod, alumnos, alumnos_nom _mod, "Alumnos.csv");
   pruebaDeCargaDeAlumnos (alumnos_cod,alumnos, alumnos_nom _mod, "PruebaAlumnos.txt");

   return 0;
}
```

NO PUEDE CAMBIAR ESTE CÓDIGO

Implemente las funciones ***cargarCursos***, ***pruebaDeCargaDeCursos***, ***cargarAlumnos*** y ***cruebaDeCargaDeAlumnos***, las funciones ***"cargar.."*** deben leer los datos del archivo ***Cursos.csv***, ***Alumnos.csv*** y ***Escalas.csv***, y colocarlos en las estructuras como se muestra en la figura No. 1, según corresponda. **Los archivos CSV deben 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.**

Las funciones ***"prueba…"*** deben emitir un reporte que pruebe, de manera clara, bien alineada y con encabezados adecuados, la carga correcta de los datos.

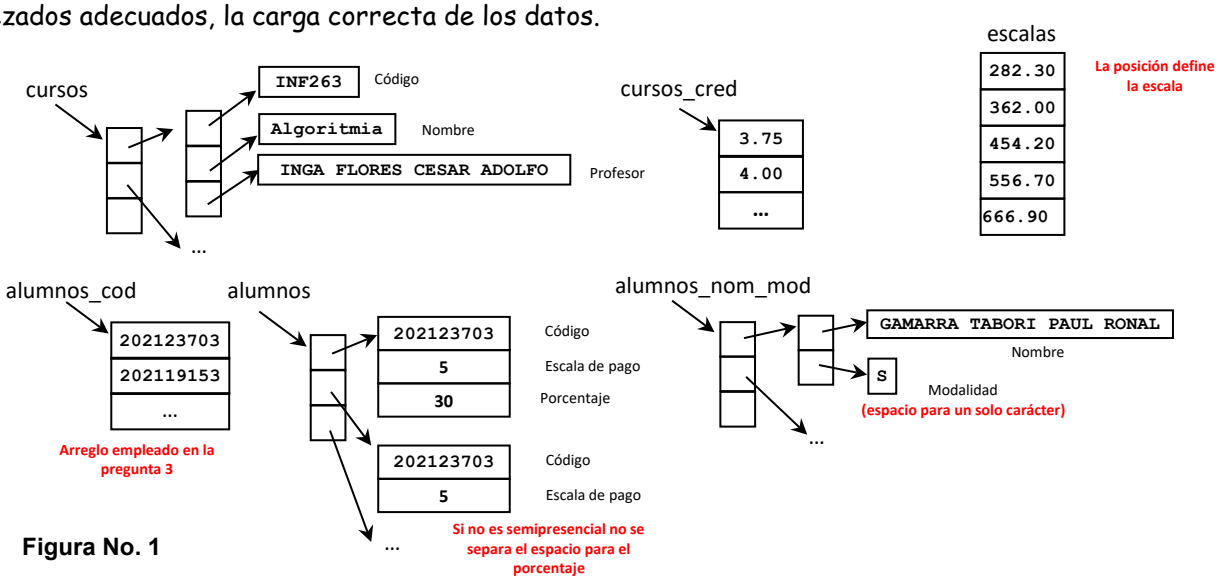

Figura No. 1

PREGUNTA 2 (7 puntos) **Este proyecto no tendrá sentido si no se desarrolla la pregunta 1**

Elabore un proyecto denominado "Pregunta02Examen01PunterosMultiples" y en él incorpore la biblioteca Preg1_Funciones_de_cursos_y_alumnos, desarrollada en la pregunta 1, y desarrolle el programa que dé solución al problema planteado. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL. **Esta pregunta no se calificará si el desarrollo lo realiza en el proyecto de la pregunta 1.**

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

```
#include "Preg1_Funciones_de_cursos_y_alumnos.h"
#include "Preg2_Funciones_de_historia_de_notas.h"

int main(int argc, char** argv) {
  char ***cursos, **alumnos_nom_mod;
   double *cursos_cred, escalas[5];
   int *alumnos_cod, **alumnos;
   cargarCursosYEscalas (cursos, cursos_cred, escalas,"Cursos.csv", "Escalas.csv");
   pruebaDeCargaDeCursos(cursos, cursos_cred,  "PruebaCursos.txt");
   cargarAlumnos (alumnos_cod, alumnos, alumnos_nom _mod, "Alumnos.csv");
   pruebaDeCargaDeAlumnos (alumnos, alumnos_int, "PruebaAlumnos.txt");int
int***cursos_nota_sem_veces;
double **costos;
char ***cursos_cod;
  cargarNotas (cursos_cod, cursos_nota_sem_veces, costos, alumnos, alumnos_nom_mod, escalas, cursos,
                cursos_cred,"HistoriaDeNotas.csv");
   pruebaDeCargaDeNotas (cursos_cod, cursos_nota_sem_veces, costos, alumnos, alumnos_nom_mod,
                "PruebaDeNotas);

   return 0;
}
```

NO PUEDE CAMBIAR ESTE CÓDIGO

Implemente las funciones ***CargarNotas*** y ***PruebaDeCargaDeNotas,*** la primera debe leer los datos del archivo ***HistoriaDeNotas.csv***, y colocarlos en las estructuras como se muestra en la figura No. 2, según corresponda (el archivo solo puede leerse una vez en todo el programa). Los espacios de memoria asignados para todos los datos deben ser DINÁMICOS Y EXACTOS. **De emplearse otro método de asignación de memoria NO se asignará puntaje en esta pregunta.**

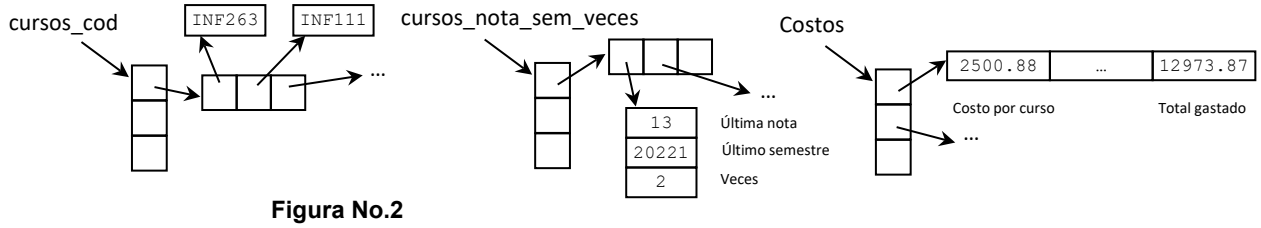

Figura No.2

La información de los datos debe estar relacionada con las posiciones de la estructura "alumnos". En la estructura "cursos_nota_sem_veces" debe colocar la nota y el semestre del último semestre que llevó el curso (pudo llevarlo varias veces). En la estructura "costos" debe acumular lo que el alumno gastó por curso, el último valor será el total gastado.

La función ***PruebaDeCargaDeNotas*** debe emitir un reporte que pruebe, de manera clara, bien alineada y con encabezados adecuados, los códigos y nombres de los alumnos, acompañados de la información cargada en las estructuras.

PREGUNTA 3 (7 puntos) **PUNTEROS GENÉRICOS**

Se pide que desarrolle un proyecto denominado "Pregunta02Examen01PuntGenericos" 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.

NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES. NO DEBE USAR ARREGLOS EXTRAS PARA ALMACENAR CONTADORES DE DATOS

Se tienen un archivo del tipo CSV, los cuales se describen a continuación:

HistoriaDeNotas.csv

```
202123703,MEC270,202302,14
202315643,MEC270,202202,13
202312000,MEC206,202301,8
……
```

Código del alumno, Código del curso, ciclo, nota

La función "main" del proyecto estará compuesto por el siguiente código:

```
#include "PunterosGenericos.h"
#include "Preg1_Funciones_de_cursos_y_alumnos.h"
#include "FuncionesAuxiliares.h"

using namespace std;

int main(int argc, char** argv) {
   int *alumnos_cod;
   void *alumnoveces;
   char **alumnos_nom_mod;
   int *alumnos_cod, **alumnos;

   cargarAlumnos (alumnos_cod, alumnos, alumnos_nom _mod, "Alumnos.csv");
   CargaCursos(alumnos_cod,alumnoveces,"HistoriaDeNotas.csv");
   ActualizaVeces(alumnoveces);
   ImprimeAlumno(alumnoveces);
   return 0;
}
```

NO PUEDE CAMBIAR ESTE CÓDIGO

DEBE USAR LA FUNCION DE LA P. 1, PERO NO ES NECESARIO HABER CARGADO TODAS LAS ESTRUCTURAS

a. **(3 puntos)** Implemente la función ***CargaCursos*** que lea los cursos en los que ha estado matriculado un alumno, desde el archivo **HistoriaDeNotas.csv**, y coloque los datos en la estructura ***alumnoveces***, representadas en la figura No. 3 según corresponda. Inicialmente las cantidades de **número de cursos aprobados, aprobados en primera, aprobados en segunda y aprobados en tercera estarán en 0**. Los espacios de memoria asignados deberán ser **dinámicos y exactos**. Utilice la función ***ImprimeAlumno***, para validar el contenido de la estructura implementada. Recuerde que el orden y los códigos de los alumnos en la estructura ***alumnosveces***, debe ser dada por la estructura ***alumnos_cod***, por tal motivo los códigos de los alumnos no se repiten. **Si un alumno no ha llevado ningún curso no debe reservarse memoria para registrar los cursos en ningún momento.**

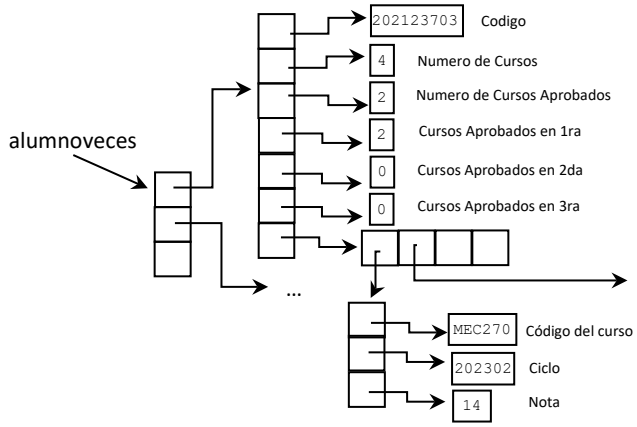

Figura No. 3

b. **(4 puntos)** Implemente la función ***ActualizaVeces***, que recorra toda la estructura ***alumnoveces***, y actualice las cantidades de **número de cursos aprobados, cursos aprobados en primera, aprobados en segunda y aprobados en tercera.** Utilice la función ***ImprimeAlumno*** para mostrar el contenido de las estructuras implementadas como reporte final de la pregunta 3.

Además, se cuenta con una biblioteca estática "***FuncionesAuxiliares***" donde está desarrollado el reporte de validación.

NOTAS IMPORTANTES:

- Se le está proporcionando una biblioteca estática para la apertura de los archivos de textos. Si lo desea puede utilizarla, pero no se le asignará puntaje por eso.
- 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**.

Profesor del curso: Rony Cueva
Miguel Guanira
Erasmo Gómez

San Miguel, 16 de mayo del 2023.