**ALGORITMOS Y ESTRUCTURAS DE DATOS** Prof. Alejandro Frankel

**TRABAJO PRÁCTICO INDIVIDUAL CODIFICADO EN C++**

**(Consideraciones, Condiciones de entrega, Corrección y Enunciado)**

**CONSIDERACIONES**

El Trabajo Práctico es un programa de complejidad empresarial con manejo de las diferentes estructuras de memoria vistas a lo largo de la materia.

**CONDICIONES GENERALES DE ENTREGA**

-Entrega individual: no puede haber dos prácticos iguales ni con el mismo juego de datos en los archivos

-Composición de la **entrega**:

-**Doc** con **carátula** (Materia, Año, Profesor, Legajo - Apellido y Nombre);

y **enunciado** del Problema +

lote de datos de prueba (los datos cargados en los archivos y los resultados de

salida esperados con esos datos)

-**archivos de datos** con los datos cargados;

-**programa en C++** que incluya (a pesar de ya entregar los archivos ya cargados):

opciones para cargar/recargar cada archivo,

opciones que permita visualizar cada archivo

opción para correr el problema.

**Fecha de entrega:**

Idealmente ***antes o con la fecha del segundo parcial***; y como máximo antes de cerrar la materia ya que es condición necesaria para aprobarla.

***La entrega y defensa oral (en los casos en que se solicite) es obligatoria***

**Recomendaciones importantes**

-no esperar hasta el último día para la entrega ya que el programa podrá contener errores lógicos no detectados;

-incluir como se pide un menú de opciones para la generación y para la visualización de cada archivo, así como para correr el ejercicio del práctico.

**Pautas para la generación de los archivos de datos originales**

Los archivos de datos deben contener suficientes registros con variantes de contenido de acuerdo a lo que debe realizar el programa (se aclara igualmente luego de los archivos el número adecuado de registros a incluir al menos).

**CORRECCIÓN DE LOS TPs**

Se verificará:

1 - Contenido de la entrega.

2 - Que no se trabaje con variables globales sino que se pase todo lo necesario a los subprogramas mediante parámetros y del tipo que corresponda (in o inout).

3 - Modularidad y funcionamiento adecuado del programa

4 - Que en caso de que consulte, conozca el desarrollo del algoritmo y funcionamiento del programa

**CONTEXTO**

El departamento de Ingeniería en Sistemas de Información mediante el SIGA obtiene tres archivos con los cuales tiene el registro de alumnos de sistemas (**Alumnos.dat**), de materias (**Materias.dat**) y de pedidos de inscripción realizadas por los alumnos (**Inscripciones.dat**) a las diferentes materias.

El departamento busca un programa en C++ que procese las inscripciones y le permita conocer datos estadísticos previos a la inscripción final de los mismos.

Los archivos contienen la siguiente información:

**ALUMNOS.DAT**

Este archivo contiene los no más de 999 alumnos registrados en ingeniería en sistemas hasta el momento sin orden. Sus campos son:

|  |  |  |
| --- | --- | --- |
| **Legajo** | **NomyAp** | **Antigüedad \*** |
| int | char[30] | int |

\*Antigüedad es la cantidad de años que el alumno lleva cursando la carrera.

**MATERIAS.DAT**

Este archivo contiene las no más de 40 materias registradas hasta el momento en ingeniería en sistemas sin orden. Sus campos son:

|  |  |  |
| --- | --- | --- |
| **CodM** | **Nombre** | **Capacidad** |
| int 6 dígitos | Char[30] | Int |

\*Capacidad es la cantidad de alumnos máxima que puedan cursarla

**INSCRIPCIONES.DAT**

Este archivo contiene los pedidos de inscripción en el orden en el que sucedieron los pedidos conteniendo:

|  |  |
| --- | --- |
| **Legajo** | **CodM** |
| int | int 6 dígitos |

***Nota:***

***A los efectos de probar el TP adecuadamente y simplificar su seguimiento, generar el archivo de ALUMNOS con al menos 6 registros (con legajos entre 1 y 6 sin orden),***

***el de MATERIAS con al menos 5 registros (con CodM entre 1 y 5 sin orden y capacidades entre 1 y 3 a efectos de probar materias que se llenen),***

***e ingresar luego por consola al menos 8 pedidos de inscripción.***

***Utilizar valores sencillos en todos los campos, por ejemplo para los alumnos 1 dígito para el legajo (4, 2, etc.) y 2 caracteres para su nombre (DD, AA, etc), y para las materias también 1 dígito para su código y 1 caracter para su nombre.***

***Si precisa crear una estructura de Materias en memoria como seguramente deba, tenga en cuenta que NO PUEDE ASOCIAR EL CÓDIGO DE MATERIA A SU POSICIÓN. Las cargará en el orden de llegada con lo cual capaz en la posición 0 queda por ejemplo la materia 5, en la 1 la 3, etc.***

***Y luego al procesar cada inscripción, para ubicar dicha materia en la estructura en memoria deberá realizar una búsqueda secuencial a partir de su código.***

Muestra de posible carga de datos (acortada) para el práctico

**ALUMNOS.DAT**

|  |  |  |
| --- | --- | --- |
| **Legajo** | **NomyAp** | **Antigüedad \*** |
| 4 | DD | 4 |
| 2 | BB | 2 |
| etc. |  |  |

**MATERIAS.DAT**

|  |  |  |
| --- | --- | --- |
| **CodM** | **Nombre** | **Capacidad** |
| 5 | E | 3 |
| 4 | D | 5 |
| etc. |  |  |

**Ejemplo de ingreso de Inscripciones por consola**

|  |  |
| --- | --- |
| Legajo | CodM |
| 5 | 2 |
| 2 | 4 |
| 1 | 3 |
| 5 | 4 |
| etc. |  |

procurar que se repitan algunos legajos y algunas materias

Se piden los siguientes reportes:

1- Cantidad de materias que no tuvieron ningún alumno inscripto

2- Por cada materia, promedio de antigüedad de los alumnos inscriptos en ella.

3- Listado en mem. dinámica ordenado por NomyAp de los alumnos que no pueden cursar una materia porque su capacidad quedó completa. Debe incluirse NomyAp y Legajo.

*(no hace falta verificar que el alumno no se repita en el listado pero si quiere hacerlo es un plus, ya que piense que un alumno podría rebotar con su inscripción en más de una materia)*.