

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. Examen

(Segundo Semestre 2015)

Indicaciones Generales:

- Duración: 3 horas.
- Se podrá usar como material de consulta sólo apuntes de clases (no fotocopias ni hojas sueltas).
- No se pueden emplear variables globales, clases, objetos (excepto cin y cout). Tampoco se podrán emplear las funciones malloc, realloc, strdup, sscanf, sprintf ni fopen.
- 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.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellas funciones que son llamadas por otras que estén incompletas. Cada módulo no debe sobrepasar las 20 líneas aproximadamente.
- La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.

Puntaje total: 20 puntos

Cuestionario:

PRIMERA PARTE:

Responda a las siguientes preguntas en un archivo de texto denominado preguntas1y2.txt. Coloque su nombre y código en el archivo.

Pregunta 1 (1.5 puntos)

El programa 1 muestra la implementación de una plantilla de funciones, el programa 2 muestra la misma implementación empleando otro estilo, explique, desde un punto de vista conceptual, por qué el primero se compila y ejecuta sin problemas mientras que el segundo no.

Programa 1	Programa 2
<pre> /* Archivo: main.cpp */ #include <iostream> using namespace std; template <typename T> void leeDatos(T*, int); int main(int argc, char** argv) { int *enteros; leeDatos(enteros, 5); return 0; } template <typename T> void leeDatos(T*arr, int n){ arr = new T[n]; for(int i=0; i<n; i++) cin>>arr[i]; } </pre>	<pre> /* Archivo: main.cpp */ #include "f.h" int main(int argc, char** argv) { int *enteros; leeDatos(enteros, 5); return 0; } /* Archivo: f.h */ #ifndef F_H #define F_H template <typename T> void leeDatos(T*, int); #endif /* F_H */ /* Archivo: f.cpp */ #include <iostream> using namespace std; template <typename T> void leeDatos(T*arr, int n){ arr = new T[n]; for(int i=0; i<n; i++) cin>>arr[i]; } </pre>

Pregunta 2 (1.5 puntos)

El programa 1 define una función para concatenar dos cadenas, el programa 2 hace lo mismo pero sobrecargando el operador +, explique, desde un punto de vista conceptual, por qué el primero se compila y ejecuta sin problemas mientras que el segundo no.

Programa 1	Programa 2
<pre>/* Archivo: main.cpp */ #include <stdio> #include <stdlib> #include <string> char * concatena(const char*c1, const char*c2){ char *aux; aux = new char[strlen(c1) + strlen(c2)+1]; strcpy(aux,c1); strcat(aux,c2); return aux; } int main(int argc, char** argv) { char n1[]="Juan", n2[]="Lopez"; char* n3; n3 = concatena(n1,n2); printf("%s\n",n3); return 0; }</pre>	<pre>/* Archivo: main.cpp */ #include <stdio> #include <stdlib> #include <string> char * operator +(const char*c1, const char*c2){ char *aux; aux = new char[strlen(c1) + strlen(c2)+1]; strcpy(aux,c1); strcat(aux,c2); return aux; } int main(int argc, char** argv) { char n1[]="Juan", n2[]="Lopez"; char *n3; n3 = n1 + n2; printf("%s\n",n3); return 0; }</pre>

SEGUNDA PARTE:

Se tienen un archivo de textos que contiene el historial de notas de los alumnos de una universidad. El archivo es similar al que se muestra a continuación:

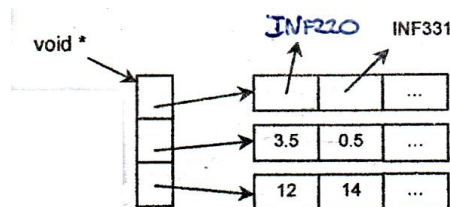
XXXX Y	← Aquí se colocará un semestre (año y semestre propiamente dicho)
López Pérez Juan	← Nombre del alumno
2008 2 2014 1	← Semestres en los que estudió
FIS281 3.5 12 INF331 0.5 14 MAT121 3.0 10 ...	← Curso, créditos y nota de los cursos que llevó en su 1er. semestre. (en este caso el 2008-2)
ING321 3.0 13 FIL166 4.5 10 ...	← Curso, créditos y nota de los cursos que llevó en su 2do. semestre (en este caso el 2009-1)
...	...
INF281 3.5 12 INF333 3.5 17
Reyes Castro María Rosa	← Nombre del otro alumno
2001 2 2007 2	...
MAT110 2.0 14 IND292 3.0 11 ...	
...	

El archivo está organizado de la siguiente manera: en la primera línea se encuentra el nombre de un alumno, en la segunda línea viene el año y semestre que ingresó seguido del año y semestre que egresó. Luego vienen tantas líneas como semestres cursó el alumno, en cada una de estas líneas aparece el código del curso, el número de créditos y la nota que obtuvo de cada curso que llevó el alumno en ese semestre. Toda esta información se repite por cada alumno.

Se pide desarrollar los siguientes proyectos:

Pregunta 3 (6 puntos)

Desarrolle una biblioteca estática de funciones denominada `almacenaSemestre`. Esta biblioteca podrá tener todas las funciones que usted crea conveniente pero tendrá dos únicas funciones que forme parte de la interfaz a la que se puede acceder desde fuera, denominadas `leerCursos` y `promedio`. La función `leerCursos` no tendrá argumentos pero devolverá un puntero genérico, esta función deberá ser capaz de leer desde la entrada estándar una línea de la forma: `FIS281 3.5 12 INF331 0.5 14 MAT121 3.0 10` y formar la siguiente estructura:



Para los arreglos secundarios se deberá emplear el método de incrementos, considerando un incremento de 5 elementos por vez (los códigos de los cursos se almacenarán en forma exacta).

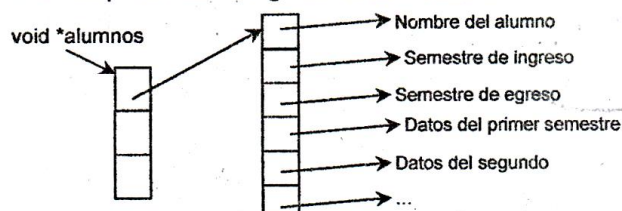
La función **promedio** deberá recibir como parámetro un puntero genérico apuntando a una estructura como la mostrada anteriormente y deberá devolver el promedio ponderado de los cursos.

Implemente con NetBeans las funciones mencionadas anteriormente en un archivo de nombre **almacenaSemestre.a** y sus correspondientes archivos **almacenaSemestre.h** y **almacenaSemestre.cpp**. Elabore un pequeño proyecto donde, de manera muy sencilla, se prueben el correcto almacenamiento de los datos y el cálculo del promedio.

Pregunta 4 (11 puntos)

Elabore un proyecto con NetBeans que permita lo siguiente (una función por ítem):

- Leer los datos de la primera línea del archivo: XXXX, Y.
- Leer los datos del archivo y formar la siguiente estructura:



Deberá emplear la biblioteca implementada en la pregunta 3. Tanto el arreglo primario como el secundario deben trabajar con memoria exacta (sin punteros NULL en ambos casos). Debe mostrar los datos almacenados de manera simple pero ordenada y clara. (6 puntos)

- Emitir un reporte en el que se muestre los promedios ponderados de los alumnos que estudiaron en el semestre (XXXX Y), con el siguiente formato:

Semestre: XXXX - Y	

Promedios ponderado obtenidos por los alumnos que cursaron ese semestre:	

Nombre	Promedio

Reyes Castro María Rosa	12.85
Guzmán Paredes Pedro	8.34
...	

(3 puntos)

- Emitir el mismo reporte anterior pero ahora ordenado por el nombre del alumno, deberá emplear la función **qsort** implementada en **cstdlib**. (2 puntos)

Al final, comprima¹ el proyecto creado en un archivo con nombre <código del alumno con 8 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Examen1\<aula>.

Profesor del curso: Miguel Guanira

San Miguel, 13 de octubre del 2015.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip). No use 7-zip o similares