

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. Examen

(Segundo Semestre 2016)

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, strtok, 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

En un archivo de textos denominado "ParteTeorica.Txt" conteste las siguientes preguntas:

Pregunta 1 (1 punto)

Dado un archivo de textos con la siguiente información:

20150105	Juan	Carlos	Perez	Gonzales
----------	------	--------	-------	----------

Y dado los siguientes códigos, re direccionando la entrada estándar para que lea el archivo:

Código 1

```
...
int codigo;
char nombre[100];

cin >> codigo;
cin.getline(nombre, 100);

cout<<codigo<<endl;
cout<<nombre<<endl;
```

Código 2

```
...
int codigo;
char nombre[100];

cin >> codigo;
cout<<codigo<<endl;

while(cin>>nombre)
    cout<<nombre<<endl;
```

Se puede apreciar que en la salida estándar se obtiene:

Código 1

20150105

Código 2

20150105
Juan
Carlos
Perez
Gonzales

Explique claramente por qué se producen estos resultados.

Pregunta 2 (1 punto)

Explique las diferencias que se producirán en una función si se declaran sus encabezados de la siguiente manera: a) `int f(int * const pt);` y b) `int f(const int * pt);`

Pregunta 3 (1 punto)

Explique las diferencias (no sintácticas), ventajas y desventajas entre una sobrecarga y una plantilla de funciones.

Pregunta 4 (1 puntos)

Dado el código mostrado a continuación, explique el por qué del resultado que nos da el programa:

```
...
void f(int p[]){
    cout << "Tam = " << sizeof(p)<<endl;
}
int main(int argc, char** argv) {
    int p[7] = {1,2,3,4,5,6,7};
    f(p);
    return 0;
}
```

SEGUNDA PARTE

Elabore los siguientes proyectos:

Pregunta 5 (6 puntos)

En el proceso de compilación de un programa se realizan una serie de tareas hasta llegar a traducir un programa fuente en uno objeto, estas tareas o etapas reciben nombres como "análisis lexical", "análisis sintáctico", "análisis semántico", etc.

El análisis lexical consiste en extraer del texto del programa grupos de caracteres con algún significado, denominados lexemas o tokens.

Se le pide elaborar una biblioteca estática de funciones denominada "AnalisisLexical", en la que se implemente una función denominada "extraeLexemas", la cual deberá recibir como parámetro una cadena de caracteres y devolver tres listas de "lexemas": una con todos los lexemas que tengan la forma de un identificador de C/C++ (palabras reservadas, variables, entre otros), otra con los valores enteros, y una última con los valores de punto flotante.

Consideraciones:

- Considerar que los lexemas no necesariamente estarán separados por espacios, tabuladores o cambios de líneas, ejemplos de eso pueden ser:
$$\text{area} = (\text{baseSup} - \text{baseInf}) * \text{altura} / 2 + 5.873;$$

¿Se compraron 65.52 ton. entre: cemento, arena, piedra y ladrillos?
- No deberá preocuparse por si hay lexemas repetidos.
- Si requiere alguna función adicional debe definirla dentro de la biblioteca pero éstas no serán accesibles a los usuarios.
- Considere que las cadenas analizadas no sobrepasarán los 500 caracteres.
- Debe desarrollar un proyecto que permita de manera sencilla probar la biblioteca.
- Tanto la biblioteca como el proyecto de prueba deben almacenarse en una carpeta denominada pregunta05, en ella deberá colocar obligatoriamente los archivos .a, .h y .cpp de la biblioteca, de faltar alguno de ellos no se calificará la pregunta y el alumno recibirá CERO como nota en esta pregunta.

Pregunta 6 (10 puntos)

Se requiere implementar un programa que permita procesar automáticamente textos provenientes de publicaciones de la red social Twitter; es decir, procesar "tweets" para guardarlos en estructuras que permitan agrupar determinadas entidades que pueden extraerse de dichos textos específicos.

En particular, en un "tweet" (que es una publicación de 140 caracteres como máximo) se puede hacer mención a otros usuarios de la red social (a partir del identificador "@", p.e.: @pucp), marcar tópicos conocidos como "hashtags" (usando el carácter "#", p.e.: #IngenieriaInformatica) o colocar enlaces o URLs (asumiremos que empiezan con "http://" o que tienen al menos un punto "." en medio del texto).

El siguiente cuadro muestra un ejemplo de una línea del archivo, donde se incluye primero el ID (numérico) del "tweet", luego el nombre de usuario del autor, y finalmente el texto de la publicación (se asume que todos los caracteres a procesar serán ASCII):

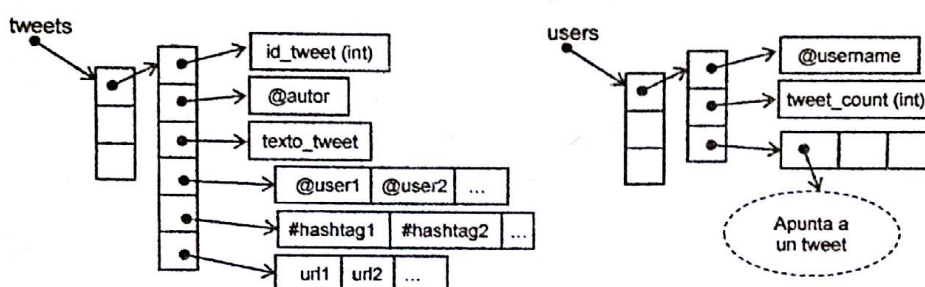
1234123 @GRPIAAPUCP La @pucp sera organizadora del evento
#CIARP2016 puede ver los detalles en http://ciarp2016.org/ o
ciarp2016.org

Por lo tanto, para manejar este archivo se le pide a usted que escriba la implementación de las funciones dadas en el siguiente programa:

```
int main(void) {  
    void * tweets; void * users;  
    leerTweets (tweets, users);  
    reporteUsuarios (users);  
    buscarUsuarioMasMencionado(...);  
    return 0;  
}
```

Parte (a): Función leerTweets

Esta función deberá leer, re-direccionando la entrada estándar de datos, los datos del archivo descrito anteriormente y guardar la información en las estructuras que se muestran a continuación:



La estructura "tweets" contiene la información del ID (como valor entero), el usuario del autor (el "@username" como cadena de caracteres), el texto completo, y tres listas de las entidades que se extraen: usuarios mencionados (cadenas de caracteres de sus "@username"), hashtags y URLs.

Por otro lado, los usuarios en la estructura "users" se irán generando a medida que vaya apareciendo un nuevo usuario en un tweet (sea como autor o como usuario mencionado). En caso sea el autor, deberá incrementar la cantidad de tweets de este usuario (tweet_count) que se almacenará como entero, y además deberá guardar dicho tweet (apuntando al elemento de la estructura "tweets") en una lista como se observa en la figura.

Parte (b): Función reporteUsuarios

La función deberá mostrar de manera clara y bien tabulada la información de los usuarios ("users"): @username, cantidad de tweets publicados (tweet_count) y lista de los IDs de dichos tweets. Esto debe hacerse usando sólo la estructura "users".

Parte (c): Función buscarUsuarioMasMencionado

Sólo se revisará esta función si las partes (a) y (b) funcionan apropiadamente.

La función deberá identificar al usuario que haya tenido más menciones en la lista de tweets. De esta forma, se deberá imprimir, a manera de validación, datos de dicho usuario como: "@username", el "tweet_count" y la cantidad de tweets donde se le ha hecho mención (puede usar las estructuras que crea conveniente, inclusive otras adicionales de apoyo si es necesario).

Al final, comprima¹ los proyectos creados 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>.

Profesores del curso: Arturo Oncevay
Miguel Guanira

San Miguel, 11 de octubre del 2016.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip).