ESTUDIOS GENERALES CIENCIAS

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

TÉCNICAS DE PROGRAMACIÓN

EXAMEN FINAL

SEMESTRE ACADÉMICO 2024-2

Horarios: Todos

Duración: 170 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación **en su mochila**.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

INDICACIONES:

- No se pueden usar apuntes de clase ni calculadoras.
- Está prohibido el acceso a Internet y a correo electrónico hasta que lo indiquen los jefes de práctica, tampoco podrá emplear dispositivos USB.
- Si se detecta omisión al punto anterior, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 30 líneas de código aproximadamente.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES. NO podrá implementar funciones en el archivo main.cpp, las funciones se deberán implementar en archivos independientes (:h y .cpp).
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- **DEBE COLOCAR SU NOMBRE Y CÓDIGO EN EL ARCHIVO main.cpp DE SU PROYECTO, DE LO CONTRARIO SE LE DESCONTARÁ 0.5 PUNTOS EN SU NOTA FINAL. NO SE HARÁN EXCEPCIONES.**
- **DEBE COLOCAR UN COMENTARIO AL INICIO DEL ARCHIVO main.cpp CON UNA DESCRIPCIÓN DE LO QUE HACE EL PROGRAMA. ESTA DESCRIPCIÓN NO DEBE SER GENÉRICA, DEBE SER ESPECÍFICA DE LO QUE HARÁ EL PROGRAMA. SE LE DESCONTARÁ 0.5 PUNTOS EN SU NOTA FINAL SI NO SE COLOCA ESTE COMENTARIO O LO QUE SE EXPRESE EN ÉL NO SEA ESPECÍFICO. NO SE HARÁN EXCEPCIONES.**
- **CADA ESTRUCTURA DEBE DEFINIRSE EN UN ARCHIVO .h INDEPENDIENTE, DE NO RESPETAR ESTE REQUERIMIENTO, NO SE ASIGNARÁ PUNTAJE EN LA DECLARACIÓN DE LAS ESTRUCTURAS.**
- **No se calificará el código puesto como comentario.**
- **En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.**
- **NO SE PERMITIRÁ LA LECTURA DE DATOS CARÁCTER POR CARÁCTER.**
- **TODA OPERACIÓN DE BÚSQUEDA DEBE REALIZARSE EN FUNCIONES INDEPENDIENTES Y DEBE CONSIDERAR QUE EL DATO BUSCADO NO SE ENCUENTRE.**

INDICACIONES INICIALES

Siga estrictamente las indicaciones que a continuación se detallan:

En la unidad de trabajo indicada por los Jefes de práctica (**si trabaja en otra unidad, no se calificará su examen y se le asignará como nota cero**), cree allí una carpeta con el nombre **"Examen2_2024_2_CO_PA_PN"** donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento se le descontará 2 puntos de la nota final). **Allí colocará los proyectos solicitados en esta prueba. NO SE HARÁN EXCEPCIONES SI NO ACATA ESTAS INDICACIONES.**

DEBE LEER TODA LA PRUEBA ANTES DE EMPEZAR A DESARROLLAR EL PROGRAMA INCLUYENDO LAS INDICACIONES Y ADVERTENCIAS

Para establecer el nuevo plan de ventas del siguiente año, un restaurante necesita hacer un análisis de las atenciones actuales para obtener la información necesaria. El área de Gestión de la Información ha brindado 4 archivos que pueden ser utilizados para el procesamiento respectivo: la lista de platos correspondiente al menú del restaurante, las atenciones realizadas, los comentarios de los clientes y un lexicón.

Desarrolle una aplicación en lenguaje C++ que realice el procesamiento correcto de los archivos brindados para que el restaurante pueda aplicar el plan indicado.

Archivo: platos.csv

```
ABC123,ANTICUCHO DE CORAZON AL PLATO,31.90
AAA123,ENSALADA COSIDA GRANDE,18.20
BBB123,POLLO A LA BRASA,25.90
CCC123,MENU NOCTURNO,20.50
…          …                …
```

Archivo: atenciones.txt

```
123456 11:25 ABC123 1 AAA123 2
111111 20:35 BBB123 1 AAA123 6 CCC123 4 DDD123 2
123321 14:12 CCC123 10
…              …                          …       …    …
```

Archivo: comentarios.csv

```
123456,Los platos estuvieron muy ricos y sabrosos, el pedido fue rapido. Volveria seguramente!!!
111111,Pesimo servicio, hubieron retrasos y el mozo estaba  chicle. Sucio.masticando
123321,Estuvo bueno, pero demoraron en atender.
123456,LA COMIDA FUE ESPANTOSA #@#&%$
 Y OLIA FEO!!!
…    …          …              …   …         …
```

Archivo: lexicon.csv

```
asco,-3
asqueroso,-2
…
bonito,3
asombroso,5
```

En el archivo "**platos.csv**" se ha colocado en cada línea el código del plato, la descripción y su precio.

En el archivo "**atenciones.txt**", se ha colocado todas las atenciones que se realizaron durante la jornada, en cada línea del archivo se ha colocado el número de atención y la hora en la que se solicitó el pedido. Luego viene la lista de platos que se solicitan, cada plato estará dado por el código del plato y por la cantidad. Cada atención puede tener un número diferente de platos.

En el archivo "**comentarios.csv**", se ha colocado todos los comentarios que se obtuvieron de los comensales durante la jornada. Por cada línea se cuenta con el número de atención y el texto del comentario hasta el fin de la línea.

En el archivo "**lexicon.csv**", se ha colocado la información de polaridad de un conjunto de palabras en el lenguaje español, la polaridad se refiere al sentimiento, positivo o negativo de una palabra. Cada línea cuenta con una palabra y como segundo elemento su polaridad. La polaridad es un valor entero en el rango [-5, 5], con este valor uno puede saber si la palabra es negativa (valores de polaridad menores a 0), positivas (valores de polaridad mayor a 0) o neutras (valor de polaridad igual a 0).

PREGUNTA 1 (10 puntos): EN ESTA PREGUNTA SERÁ OBLIGATORIO EL USO DE ARREGLOS

Cree un proyecto en NetBeans con el nombre: "Pregunta1_2024_2" (de no respetar este nombre se le descontarán dos puntos de su nota final – NO SE HARÁN EXCEPCIONES) y en él desarrolle el programa que resuelva el problema que se describe a continuación. El proyecto debe estar dentro de la carpeta creada previamente y debe contener todas las funciones necesarias para su correcto funcionamiento.

CONSIDERACIONES

a) NO DEBE HACER QUE DOS O MÁS PUNTEROS APUNTEN AL MISMO DATO.
b) No debe repetir código innecesariamente, tareas como la lectura y escritura de los textos deben ser realizadas en una única función que se adapte a todas las situaciones del problema.
c) Será parte importante de la nota el formato del reporte, éste deberá ser lo más parecido a la muestra dada. En este sentido, todos los valores deben estar correctamente alineados. No se podrá emplear el carácter de tabulación ('\t') para la emisión del reporte.

Las tareas por realizar son las siguientes, las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente:

a) (1.0 punto) Definir las siguientes estructuras:

- **Palabra**: que debe contener los siguientes campos: 1) **texto** (cadena de caracteres dinámica) y 2) **polaridad** (valor entero).
- **Comentario**: que debe contener los siguientes campos: 1) **texto** (cadena de caracteres dinámica), 2) **pre_procesado** (cadena de caracteres dinámica), 3) **palabras** (arreglo estático de Palabra, considerar 20 palabras máximo), 4) **cantidad_palabras** (valor entero) y 5) **polaridad_total** (valor entero).
- **Plato:** que debe contener los siguientes campos: 1) **codigo** (cadena de caracteres dinámica), 2) **nombre** (cadena de caracteres dinámica), 3) precio (valor de punto flotante) y 4) **cantidad** (valor entero).
- **Atencion**: que debe contener los siguientes campos: 1) **codigo** (valor entero), 2) **platos_atendidos** (arreglo dinámico de estructura Plato), 3) **cantidad_platos** (valor entero), 4) **total_venta** (valor de punto flotante), 5) **hora** (valor entero), 6) **comentarios** (arreglo dinámico de estructura Comentario), 7) **cantidad_comentarios** (valor entero). Considerar máximo de 20elemtos en los arreglos.

b) (1.0 punto) Leer los datos del archivo lexicon.csv y alojarlos en una variable llamada **lexicon** que es un arreglo estático de **Palabra** de no más de 100 palabras.

c) (1.0 punto)Leer los datos del archivo platos.csv y alojarlos en una variable **platos** del tipo arreglo dinámico de **Plato** de no más de 200 platos.

d) (3.0 puntos) Leer los datos del archivo **atenciones.csv** y alojarlos en una variable llamada **atenciones** que es en un arreglo estático de **Atencion** de no más de 100 atenciones. Para llenar el campo **platos_atendidos** deberá hacer uso del del arreglo **platos** para buscar el plato que se ha vendido. Con esta información y las cantidades, puede calcular el valor del campo **total_venta**.

e) (2.5 puntos) Leer los datos del archivo **comentarios.csv** y haciendo uso de los arreglos **atenciones** y **lexicon**, realizar el siguiente proceso por cada comentario.
   - *Pre procesar el comentario*: esto consiste en remover todos los caracteres NO alfanuméricos del comentario, con excepción del espacio. Pasar todas las letras a minúsculas. El resultado de este proceso debe alojarlo en el campo **pre_procesado** del arreglo **atenciones**. Para realizar el pre procesamiento se le recomienda usar las funciones **isalpha** y **tolower (ver ayuda al final de la prueba).**
   - *Procesar el comentario*: dividir el comentario en palabras del tipo **Palabra** y completar su información, para calcular la polaridad de cada palabra debe buscar en el **lexicon** y obtener la información de la polaridad, de no encontrarla el valor devuelto debe ser 0. Estas palabras ya completas deben ser almacenadas en el campo **palabras.**
   - *Polaridad total*: es la suma de las polaridades de cada palabra en el comentario.

f) (1.5 puntos) Emitir el siguiente reporte:
ReporteAtencionesAnalisisSentimiento.txt

```
                              EMPRESA DE REPARTOS A DOMICILIO TP S. A.
                                       RELACIÓN DE ATENCIONES
=============================================================================================================
No.  ATENCION 123456                                                        Atendido a las: 11:45 AM
-------------------------------------------------------------------------------------------------------------
ABC123) ANTICUCHO DE CORAZON AL PLATO         31.90      1     31.90
AAA123) ENSALADA COSIDA GRANDE                18.20      2     36.40
…    …      …                                      …         …         …
Polaridad total de los comentarios :   24
=============================================================================================================
No.  ATENCION 111111                                                        Atendido a las: 20:35 AM
…
=============================================================================================================
```

PREGUNTA 2 (10 puntos): EN ESTA PREGUNTA SERÁ OBLIGATORIO EL USO ARCHIVOS BINARIOS

Cree un proyecto en NetBeans con el nombre: "Pregunta2_2024_2" (de no respetar este nombre se le descontarán dos puntos de su nota final – NO SE HARÁN EXCEPCIONES) y en él desarrolle el programa

que resuelva el problema que se describe a continuación. El proyecto debe estar dentro de la carpeta creada previamente y debe contener todas las funciones necesarias para su correcto funcionamiento.

CONSIDERACIONES

- No puede desarrollar una tarea si no ha desarrollado por completo la tarea anterior, de hacerlo no se corregirá esa tarea.
- Será parte importante de la nota el formato del reporte, éste deberá ser lo más parecido a la muestra dada. En este sentido, todos los valores deben estar correctamente alineados. No se podrá emplear el carácter de tabulación ('\t') para la emisión del reporte.
- Toda operación de búsqueda debe realizarse en una función independiente. No se considerará en la calificación los procesos de búsqueda que estén contenidos en el código de otro proceso. Las funciones de búsqueda deben considerar la posibilidad que el dato buscado no se encuentre.

Las tareas por realizar son las siguientes, las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente:

a) (1.0 punto) Definir las siguientes estructuras:
   - **Palabra**: que debe contener los siguientes campos: 1) **texto** (cadena de caracteres estática) y 2) **polaridad** (valor entero).
   - **Comentario**: que debe contener los siguientes campos: 1) **texto** (cadena de caracteres estática), 2) **pre_procesado** (cadena de caracteres estática), 3) **palabras** (arreglo de Palabra), 4) **cantidad_palabras** (valor entero) y 5) **polaridad_total** (valor entero). Un comentario puede tener máximo 20 palabras.
   - **Plato:** que debe contener los siguientes campos: 1) **codigo** (cadena de caracteres estática), 2) **nombre** (cadena de caracteres estática) y 3) **precio** (valor de punto flotante)
   - **Atencion**: que debe contener los siguientes campos: 1) **codigo** (valor entero), 2) **total_venta** (valor de punto flotante), 3) **hora** (valor entero), 4) **comentarios** (arreglo estático de estructura Comentario) y 5) **cantidad_comentarios** (valor entero). En una atención como máximo se pueden tener 20 comentarios.

b) (1.0 punto) Leer los datos del archivo lexicon.csv y colocarlos en un archivo binario llamado **lexicon.bin**. Genere un reporte previo llamado **"ReporteLexicon.txt"** , el reporte debe tener título, nombres en las columnas y los datos con el formato adecuado.

c) (1.0 punto) Leer los datos del archivo platos.csv y colocarlos en un archivo binario llamado **platos.bin**. Genere un reporte llamado "**ReportePruebaPlatos.txt**", el reporte debe tener título, nombres en las columnas y los datos con el formato adecuado.

d) (1.5 punto) Leer los datos del archivo atenciones.csv y colocarlos en un archivo binario llamado **atenciones.bin,** debe utilizar el archivo **platos.bin** para actualizar el campo totalVenta, que es la suma acumulada del precio del plato por la cantidad. Recuerde que los campos que no se lean del archivo, deben llenarse con valores adecuados.

e) (1.5 punto) Emitir un reporte "**ReporteInicialAtenciones.txt**" que muestre la información del archivo **atenciones.bin**, el reporte debe estar desarrollado para mostrar todos los datos de las estructuras que contiene cada registro del archivo. El reporte debe tener título, nombres en las columnas y los datos con el formato adecuado.

f) (3.5 puntos) Leer los datos del archivo **comentarios.csv** y con el archivo binario **lexicon.bin** actualice la información del archivo **atenciones.bin**. realizar el siguiente proceso por cada comentario
   - *Pre procesar el comentario*: esto consiste en remover todos los caracteres NO alfanuméricos del comentario, con excepción del espacio. Pasar todas las letras a minúsculas. El resultado de este proceso debe alojarlo en el campo **pre_procesado** del archivo **atenciones.bin**. Para realizar el pre procesamiento se le recomienda usar las funciones **isalpha** y **tolower (ver ayuda al final de la prueba).**
   - *Procesar el comentario*: dividir el comentario en palabras del tipo **Palabra** y completar su información, para calcular la polaridad de cada palabra debe buscar en el archivo **lexicón.bin** y obtener la información de la polaridad, de no encontrarla el valor devuelto debe ser 0. Estas palabras ya completas deben ser almacenadas en el campo **palabras.**
   - *Polaridad total*: es la suma de las polaridades de cada palabra en el comentario.

g) (0.5 puntos) Emitir nuevamente el reporte de atenciones (de la parte e) esta vez con el nombre "**ReporteActualizadoAtenciones.txt**".

Al finalizar el examen, comprima la carpeta que contiene los dos proyectos 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 y súbalo a la tarea programada en PAIDEIA para este examen.

AYUDA EN EL USO DE LAS FUNCIONES ISALPHA Y TOLOWER:

```
#include <iostream>
#include <cstring>
using namespace std;

int main(int argc, char** argv) {
   char linea[100] = "Hi!!! ESTE es el uso de isalpha() y tolower()";
   char linea_pre_procesada[100], c;
   int k = 0;
   for (int i = 0; linea[i]; i++) {
      c = linea[i];
      if(isalpha(c) or c == ' '){
         linea_pre_procesada[k] = tolower(c);
         k++;
      }
   }
   linea_pre_procesada[k]=0;
   cout<<linea_pre_procesada<<endl;
   return 0;
}

Resultado:
hi este es el uso de isalpha y tolower
```

ADVERTENCIAS:
Obligatoriamente debe desarrollar sus proyectos bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo.

CRITERIOS DE CALIFICACIÓN:
1. Si el programa entregado presenta más de tres errores de sintaxis serán calificados sobre la mitad del puntaje. 2. Si el programa no muestra los resultados o los muestren y no sean correctos, no podrán tener más del 75% de la nota. 3. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido. Las variables deben empezar con una minúscula, se emplearán mayúsculas o guiones para separar las palabras compuestas (p. e.: baseInferior). 4. Se descontará 15% de la nota si no se colocan comentarios relevantes, 5. No se calificará el código puesto como comentario, ni aquellas funciones que no son invocadas en el proyecto. 6. No se calificarán aquellas funciones implementadas en el archivo main.cpp

San Miguel, 02 de diciembre del 2024