

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. Examen

(Segundo Semestre 2019)

Indicaciones Generales:

- Duración: 3 horas.
- Se podrá usar como material de consulta sólo apuntes de clases (NO fotocopias, impresos ni hojas sueltas).
- No se pueden emplear variables globales, estructuras, ni objetos (con excepción de los elementos de `iostream`, `iomanip` y `fstream`). 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.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada módulo **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 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**.
- **NO SE CALIFICARÁN AQUELLAS FUNCIONES DESARROLLADAS EN EL MISMO ARCHIVO QUE LA FUNCIÓN `main`.**
- El código comentado **NO SE CALIFICARÁ**.
- 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%.
- **La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

Puntaje total: 20 puntos

Cuestionario:

PARTE 1 – PUNTEROS GENÉRICOS

Una empresa productora ha decidido establecer un almacén de consignación para sus clientes. Este tipo de almacenes permite a los clientes minoristas vender la mercadería, que se encuentra en este lugar y pagar **solo** por lo vendido a fin de mes, a este proceso de pago se le denomina **liquidación** del mes. Por tal motivo se desea elaborar una aplicación que permita realizar esta liquidación para cada cliente en un determinado mes. Para esta operación se cuenta con los archivos de ventas realizadas por los clientes, precios vigentes de los productos y nombres de los clientes, los tres archivos tiene formato CSV, y la información se guarda como se muestra a continuación:

ventas.csv
20190208,20015550,A106001,20
20190209,20189890,A106002,30
...
productos.csv
A106001,Arroz 1 kg,3.5
A106002,Azucar Rubia 1 kg,2.5
...
clientes.csv
20015550,La bodeguita de Don Pepe
20189890,La preferida
...

En el primer archivo se encuentran las ventas realizadas por los clientes, donde aparece la fecha de la operación, el RUC del cliente, código del producto y cantidad de productos vendidos. En el segundo archivo se han registrado los datos de los productos, por tal motivo en cada línea se indica el código del producto, nombre y el precio unitario. En el tercer archivo se tiene un listado de todos los clientes que pueden vender utilizando el almacén de consignación, por lo cual en cada línea aparece el RUC del cliente minorista y su nombre o razón social.

La aplicación la desarrollará en un proyecto denominado "**Pregunta01 - Examen01**", en donde el archivo `main.cpp` contenga el siguiente código:

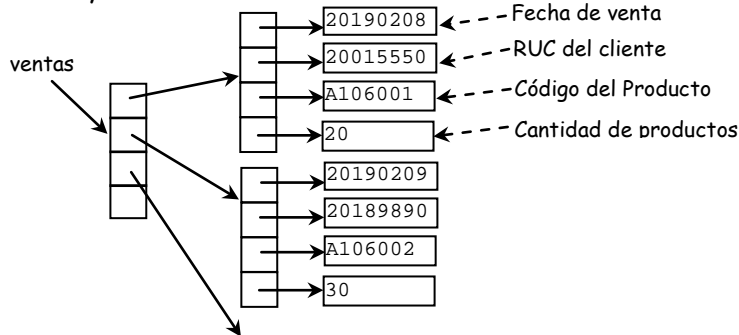
```
#include "FuncionesParte01Ex1.h"
int main(void) {
    void *ventas, *productos, *clientes;

    leeVentas(ventas);
    imprimeVentas(ventas);
    leeProductos(productos);
    imprimeProductos(productos);
    leerClientes(clientes);
    imprimeClientes(clientes);
    liquidaVentas(clientes,ventas,productos);
    return 0;
}
```

NO PUEDE
MODIFICAR
ESTE CÓDIGO

Pregunta 1 (2 puntos)

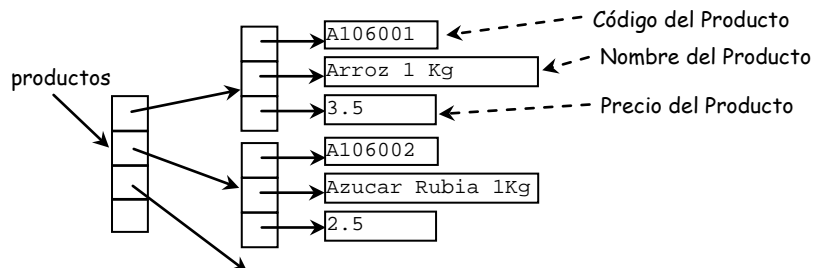
Implemente las funciones "**leeVentas**" e "**imprimeVentas**", la primera función deberá leer los datos del archivo "ventas.csv" y colocar la información en una estructura como se muestran a continuación:



Los espacios de memoria asignados deberán ser dinámicos y **exactos**. La función **imprimeVentas** tiene la finalidad de verificar que los datos hayan sido correctamente colocados en la estructura, los datos deben aparecer muy bien tabulados de manera que cualquiera pueda entender lo que aparece en el reporte.

Pregunta 2 (1.5 puntos)

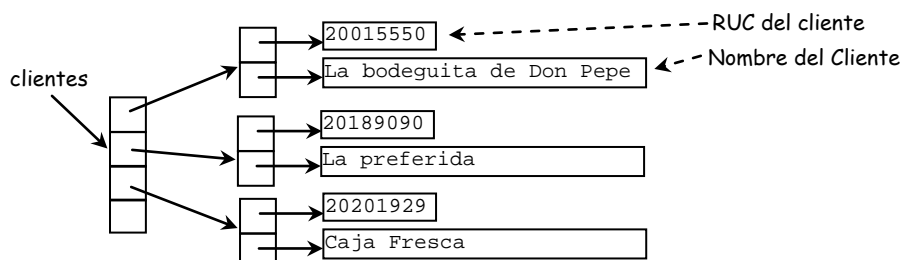
Implemente las funciones "**leeProductos**" e "**imprimeProductos**", la primera función deberá leer los datos del archivo "productos.csv" y colocar la información en una estructura como se muestra a continuación:



Los espacios de memoria asignados deberán ser dinámicos y **exactos**. La función **imprimeProductos** tiene la finalidad de verificar que los datos hayan sido correctamente colocados en la estructura, los datos deben aparecer muy bien tabulados de manera que cualquiera pueda entender lo que aparece en el reporte.

Pregunta 3 (1.5 puntos)

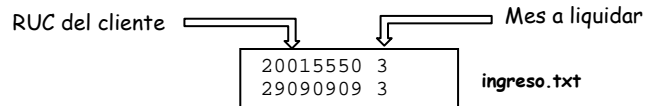
Implemente las funciones "**leeClientes**" e "**imprimeClientes**", la primera función deberá leer los datos del archivo "clientes.csv" y colocar la información en una estructura como se muestran a continuación:



Los espacios de memoria asignados deberán ser dinámicos y **exactos**. La función **imprimeClientes** tiene la finalidad de verificar que los datos hayan sido correctamente colocados en la estructura, los datos deben aparecer muy bien tabulados de manera que cualquiera pueda entender lo que aparece en el reporte.

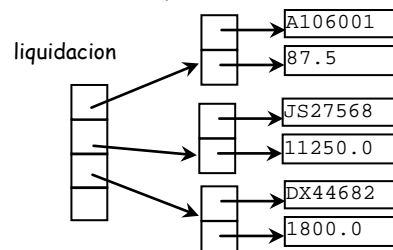
Pregunta 4 (5 puntos)

Implemente la función "**liquidaVentas**", esta función recibe un archivo con el RUC del cliente y el mes a liquidar las ventas, similar al siguiente formato:



Este archivo puede tener el RUC de uno o más clientes. El proceso de liquidación de las ventas de un cliente tiene los siguientes pasos:

- Leer del archivo **ingreso.txt** el RUC del cliente y mes a liquidar.
- Generar la estructura **liquidacion** que almacena la venta total por cada producto vendido por el cliente, durante el mes solicitado. El monto vendido debe acumularse por producto, recorriendo la estructura **ventas**, considerando el precio de la estructura **productos**. La estructura tiene el siguiente formato:



- Una vez se tenga cargada todas las ventas de un cliente de acuerdo al mes correspondiente, se debe generar un archivo con la liquidación solicitada, de la siguiente forma:

Liquidación: La bodeguita de Don Pepe		
CODIGO	NOMBRE DEL PRODUCTO	IMPORTE
A106001	Arroz 1 Kg	87.5
JS27568	Cocina a gas meza vitroceramica FFEF 3013	11250.0
DX44682	Tostadora de pan con 4 slots 5632	1800.0

El nombre del archivo debe ser el RUC de la empresa concatenado con el mes, por ejemplo: 200155503.txt

- Finalmente se debe continuar con el siguiente cliente del archivo ingreso.txt

PARTE 2 - PUNTEROS MÚLTIPLES

Se desea elaborar una aplicación que permita gestionar las infracciones de tránsito cometidas por los diferentes choferes del país. Para realizar esta labor se cuenta con tres archivos CSV, los que se muestran a continuación:

Archivo 1	Archivo 2
63736112,ZAMORA/ZVALETA/RONAL MANUEL,989320773,K0D-676,T9A-930 45043076,VEGA/VILCARA/CARMELA TERESA,986528258,S7E-946,I1D-713,R1T-735 ...	81485316,J5T-691,128 25518120,E1J-798,118 ...
Archivo 3	
103,Detener el vehículo bruscamente sin motivo.,Grave,316.00 201,Dejar mal estacionado el vehículo en lugares permitidos.,Leve,158.00 ...	

En el primer archivo se encuentra la lista de conductores del país, donde aparece el DNI, nombre y teléfono del conductor, seguido por los códigos de las placas de los autos que posee. En el segundo archivo se han registrado las infracciones de tránsito que se han cometido en un período de tiempo, donde en cada línea se indica el DNI y la placa del conductor que cometió la falta, seguido del código de la infracción que cometió. En el tercer archivo se tiene un listado de todas las infracciones que se pueden cometer en el país, donde en cada línea aparece el código de la infracción, seguido de su descripción, luego viene la gravedad de la infracción y finalmente la multa por haberla cometido.

La aplicación la desarrollará en un proyecto denominado "**Pregunta02-Examen01**", en donde el archivo main.cpp contenga el siguiente código:

```
#include "FuncionesParte02Ex1.h"
int main(void) {
    int *dni;
    char **conductor, ***placas;
    double *multas, ***montosXMultas;

    leerConductores(dni, conductor, placas);
    imprimirConductores(dni, conductor, placas);

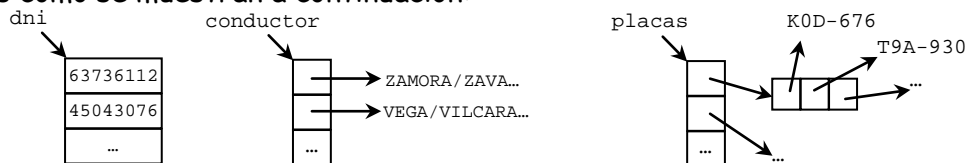
    leerMultas(multas);
    imprimirMultas(multas);

    leerInfraccionesCometidas(dni, placas, multas, montosXMultas);
    imprimirInfracCometidas(dni, conductor, placas, montosXMultas);
    return 0;
}
```

**NO PUEDE
MODIFICAR
ESTE CÓDIGO**

Pregunta 5 (4 puntos)

Implemente las funciones **"leerConductores"** e **"imprimirConductores"**, la primera función deberá leer los datos de un archivo similar al indicado como "Archivo 1" y colocar la información en estructuras como se muestran a continuación:



La asignación de memoria debe ser por incrementos de 5 en 5 en todos los casos, salvo en las cadenas que deben ser exactas.

La función **imprimirConductores** deberá mostrar en un archivo de textos el contenido de las estructuras como se muestra a continuación (no podrá emplear el carácter '\t').

REGISTRO DE CONDUCTORES		
DNI	CONDUCTOR	AUTOS
1) 63736112	ZAMORA/ZAVALETA/RONAL MANUEL	1) K0D-676 2) T9A-930
2) 45043076	VEGA/VILCARA/CARMELA TERESA	1) S7E-946
...

Los nombres de ambos archivos de texto deben ser ingresados por teclado.

Pregunta 6 (2 puntos)

Implemente las funciones **"leerMultas"** e **"imprimirMultas"**, la primera función deberá leer los datos de un archivo similar al indicado como "Archivo 3" y colocar en el arreglo **"multas"** el monto de la multa que se pueden cometer en el país, el arreglo debe poder ser manipulado como una **"tabla Hash"** de modo que se pueda rápidamente el monto de una multa determinada. La asignación de memoria deberá ser óptima.

La función **imprimirMultas** deberá mostrar en un archivo de textos el contenido de las estructuras como se muestra a continuación (no podrá emplear el carácter '\t').

MULTAS POR INFRACCIONES DE TRÁNSITO		
	CODIGO	MULTA
1)	103	316.00
2)	201	158.00
...

En el reporte solo a deben aparecer las multas existentes. Los nombres de ambos archivos de texto deben ser ingresados por teclado.

Pregunta 7 (4 puntos)

Implemente las funciones **"leerInfraccionesCometidas"** e **"imprimirInfracCometidas"**, la primera función deberá leer los datos de un archivo similar al indicado como "Archivo 2" y colocar la información en la estructura como se muestran a continuación:



La estructura "**montosXMultas**" debe contener el acumulado de todas las multas que cometió cada conductor por cada vehículo que posee ^(*), y el total de sus multas ^(**). Los espacios de memoria deben ser exactos.

La función **imprimirInfracCometidas** deberá mostrar en un archivo de textos el contenido de las estructuras como se muestra a continuación (no podrá emplear el carácter '\t').

REGISTRO DE CONDUCTORES			
DNI	CONDUCTOR	AUTOS	Multas
1) 63736112	ZAMORA/ZAVALETA/RONAL MANUEL	1) K0D-676	1543.89
		2) T9A-930	845.25
		Total:	2389.14
2) 45043076	VEGA/VILCARA/CARMELA TERESA	1) S7E-946	...
...

Los nombres de ambos archivos de texto deben ser ingresados por teclado.

CONSIDERACIONES FINALES:

- Cree en el computador una carpeta de trabajo con la siguiente ruta: c:\temp\Examen1. En ella colocará los proyectos que den solución a los problemas planteados.
- 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.
- De no respetarse el nombre de los proyectos se descontará 1 punto por cada trasgresión.
- Cualquier trasgresión de las indicaciones dadas en la prueba ameritará un descuento de puntaje por cada una de ellas.

NO SE HARÁN EXCEPCIONES

Al finalizar el examen, comprima¹ la carpeta **Examen1** 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\<código del horario>\<aula>. El acceso a la Intranet quedará cerrado automáticamente a las 11:05 am. por lo que el alumno que no suba alguno de los proyectos a la Intranet recibirá como nota CERO en esa pregunta. **NO SE HARÁN EXCEPCIONES.**

Profesor del curso: Rony Cueva
Miguel Guanira

San Miguel, 15 de octubre del 2019.

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 7z**.