

TÉCNICAS DE PROGRAMACIÓN

SEGUNDO EXAMEN

SEMESTRE ACADÉMICO 2021-2

Horario: Todos

Duración: 170 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:

- SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO, Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .c O .h CON FECHA U HORA DE CREACIÓN ANTERIOR AL EXAMEN SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

INDICACIONES:

- 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. CADA ESTRUCTURA DEBE DEFINIRSE EN UN ARCHIVO .h INDEPENDIENTE.**
- NO SE PUEDEN EMPLEAR ARCHIVOS AUXILIARES, NI VARIABLES GLOBALES.
- 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 CADA UNO DE LOS ARCHIVOS .h Y .c QUE EMPLEE EN SUS PROYECTOS.**

Cree una carpeta con el nombre de **“TP20212_Examen_Final”** (de no respetar este nombre se le descontarán dos (2) puntos de su nota final – NO SE HARÁN EXCEPCIONES) y en ella coloque los dos proyectos que den solución a las preguntas planteadas en este examen. Es obligatorio que cada pregunta sea desarrollada en un proyecto independiente, de no acatar esta indicación solo se calificará una pregunta.

Una tienda de ventas de artículos por Internet quiere implementar aplicaciones que le permitan procesar los pedidos de sus clientes. Los clientes registrados podrán realizar los pedidos de los productos que desean comprar a través de la página web de la tienda, luego la tienda les llevará los productos pedidos a sus domicilios. Para realizar esta labor, se cuentan con los siguientes archivos:

StockDeProductos.csv			
VG6534,	Lapicero Rojo F14,	70,	3.45
AN6791,	Hojas bond 100u.,	32,	15.28
...			

Cada línea del archivo contiene el código, descripción, cantidad en stock y precio de un producto. El archivo no está ordenado.

Pedidos.txt			
13/10/2021	10:14:25	564321	67213455
7/9/2021	14:22:17	769812	71990230
...			

Cada línea del archivo contiene la fecha, hora, número de pedido y DNI del cliente que lo realizó. No se repite el número de pedido en el archivo. El archivo no está ordenado.

DetalleDeLosPedidos.txt		
112233	SV7123	10
673212	AR1090	23
112233	VG8237	3
...		

Cada línea del archivo contiene el número de pedido, el código del producto y la cantidad solicitada. Los números de pedidos y los códigos de los productos pueden aparecer varias veces en el archivo, esto porque un pedido puede estar compuesto por varios productos solicitados y un mismo producto puede estar solicitado en varios pedidos. El archivo no está ordenado. Los productos de un mismo pedido no necesariamente estarán uno a continuación del otro.

Pregunta 01 (10 puntos) - ARCHIVOS BINARIOS

EN ESTA PREGUNTA NO PODRÁ COLOCAR LOS DATOS EN ARREGLOS, SALVO POR EL CAMPO DESCRITO EN LA ESTRUCTURA QUE SE DETALLA MÁS ADELANTE Y EL ARREGLO DE CADENA DE CARACTERES QUE SE UTILIZA EN LA FUNCIÓN SACAPALABRAS.

Cree un proyecto en NetBeans con el nombre: **“TP_ArchBin_Pregunta_01”** (de no respetar este nombre se le descontarán dos (2) puntos de la pregunta – NO SE HARÁN EXCEPCIONES) y en él desarrolle el programa que resuelva el siguiente problema con las tareas que se describen a continuación.

- (1.5 punto) Cree un archivo binario con el contenido del archivo **“StockDeProductos.csv”**. El archivo se denominará **“StockDeProductos.bin”** y debe contener los siguientes campos: **codigo** (cadena de caracteres), **descripcion** (cadena de caracteres), **stock** (valor entero) y **precio** (valor de punto flotante).
En un reporte muy simple, debe verificar la creación correcta del archivo, esta verificación debe hacerse en una función independiente y debe ser ejecutada fuera de la función que crea el archivo. El reporte debe salir en un archivo de texto.
- (2.5 puntos) Cree un archivo binario denominado **“PedidosPorInternet.bin”** que contenga los siguientes campos: **numero** (entero), **fecha** (valor entero), **hora** (valor entero), **cliente** (valor entero con el DNI del cliente), **productosPedidos** (arreglo de estructuras tipo **ProductoSolicitado**), **cantProdPed** (entero). La estructura **ProductoSolicitado** debe contener los siguientes campos: **codigo** (cadena de caracteres), **descripcion** (cadena de caracteres), **cantidad** (valor entero), **precioUnitario** (valor de punto flotante) y **subtotal** (valor de punto flotante). Considerar que un pedido puede tener hasta 20 artículos.
Inicialmente el archivo debe llenarse con todos los datos del archivo **Pedidos.txt**, el resto de campos no tendrán valores definidos salvo aquellos que se requieran inicializar.
En un reporte muy simple, debe verificar la creación correcta del archivo (no requiere aquí disgregar las fechas y horas), esta verificación debe hacerse en una función independiente y debe ser ejecutada fuera de la función que crea el archivo. El reporte debe salir en un archivo de texto.
- (4 puntos) Empleando los archivos **DetalleDeLosPedidos.txt** y **“StockDeProductos.bin”** debe completar los campos correspondientes del archivo **“PedidosPorInternet.bin”**. Si el stock del producto pedido es mayor o igual a la cantidad solicitada se colocará el pedido en el archivo con la cantidad solicitada y se actualizará el stock del producto en el archivo **“StockDeProductos.bin”**, si hay stock del producto pero es menor a lo solicitado se debe registrar el pedido pero solo con la cantidad que hay en el stock y debe actualizar el stock. Si no hay stock el pedido del producto no se registra.
- (2 puntos) Emita el siguiente reporte:

ReportedeDePedidos.txt

=====				
TIENDA VIRTUAL LA MAGNIFICA				
ATENCIÓN DE PEDIDOS				
=====				
Pedido No. 564321	Fecha:	13/10/2021	Hora:	10:14:25
Cliente:				
DNI: 67213455				
Productos:				
Codigo	Descripcion	Cantidad	Precio Unitario	Subtotal
NR60310	Pincel No. 2	2	15.25	30.50
...
=====				
Pedido No. ...	Fecha:	...	Hora:	...
...				

Consideraciones:

- Los archivos CSV y TXT solo se pueden leer una vez.
- 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 descritas deben realizarse en el orden indicado, no se calificarán las tareas desarrolladas si las previas no se desarrollaron.
- En la calificación se tomará en cuenta el uso correcto de las funciones para leer las cadenas de caracteres, el buen uso de los punteros y de la memoria dinámica asignada, deberá liberar los espacios que ya no se usen, no deberá dejar espacios de memoria sueltos, tampoco deberá gastar memoria dinámica innecesariamente.

Pregunta 02 (10 puntos) - ESTRUCTURAS Y ARREGLOS

EN ESTA PREGUNTA NO PODRÁ EMPLEAR LOS ARCHIVOS BINARIOS DESARROLLADOS EN LA PREGUNTA ANTERIOR NI OTRO ARCHIVO AUXILIAR.

Cree un proyecto en NetBeans con el nombre: **“TP_Struct_Pregunta_02”** (de no respetar este nombre se le descontarán dos (2) puntos de la pregunta – NO SE HARÁN EXCEPCIONES) y en él desarrolle el programa que resuelva el siguiente problema con las tareas que se describen a continuación.

- a) (1.5 punto) Defina en un arreglo dinámico de estructuras de tipo **StockProd** (**struct StockProd ***), el arreglo se denominará **productos**. La estructura debe contener los siguientes campos: **codigo** (char *), **descripcion** (char *), **stock** (valor entero) y **precio** (valor de punto flotante). Llene el arreglo con la información del archivo **“StockDeProductos.csv”**.

En un reporte muy simple, debe verificar la creación correcta del arreglo, esta verificación debe hacerse en una función independiente y debe ser ejecutada fuera de la función que crea el archivo. El reporte debe salir en un archivo de textos.

- b) (2.5 puntos) Defina en un arreglo dinámico de estructuras de tipo **stPedidos** (**struct stPedidos ***) el arreglo se denominará **pedidos**. La estructura debe contener los siguientes campos: **numero** (entero), **fecha** (valor entero), **hora** (valor entero), **cliente** (valor entero con el DNI del cliente), **productosEntregados** (arreglo dinámico de estructuras tipo **Entregado – struct Entregado ***), **cantProdEnt** (entero), **productosNoEntregados** (arreglo dinámico de estructuras tipo **NoEntregado – struct NoEntregado ***), **cantProdNoEnt** (entero). La estructura **Entregado** debe contener los siguientes campos: **codigo** (char *), **descripcion** (char *), **cantEntregada** (valor entero), **precioUnitario** (valor de punto flotante) y **subtotal** (valor de punto flotante). La estructura **NoEntregado** debe contener los siguientes campos: **codigo** (char *), **descripcion** (char *), **cantNoEntregada** (valor entero).

Llene los campos correspondientes con el archivo **Pedidos.txt**, el resto de campos no tendrán valores definidos salvo aquellos que se requieran inicializar. Considerar que un pedido puede tener hasta 20 artículos.

En un reporte muy simple, debe verificar la creación correcta del arreglo (no requiere aquí disgregar las fechas y horas), esta verificación debe hacerse en una función independiente y debe ser ejecutada fuera de la función que crea el archivo. El reporte debe salir en un archivo de texto.

- c) (4 puntos) Empleando el archivo **DetalleDeLosPedidos.txt** debe completar los campos correspondientes del arreglo **Pedidos**. Si el stock del producto pedido es mayor o igual a la cantidad solicitada, se colocará el pedido en el arreglo **productosEntregados** con la cantidad solicitada y actualizar el stock del producto en el arreglo **Productos**, si hay stock del producto pero es menor a lo solicitado se debe registrar el pedido pero solo con la cantidad que hay en el stock, la cantidad no entregada debe registrarse en el arreglo **productosNoEntregados** y debe actualizar el stock. Si no hay stock el pedido se registra en el arreglo **productosNoEntregados**.

- d) (2 puntos) Emita el siguiente reporte:

ReportedeDePedidos.txt

=====				
TIENDA VIRTUAL LA MAGNIFICA				
ATENCIÓN DE PEDIDOS				
=====				
Pedido No. 564321	Fecha:	13/10/2021	Hora:	10:14:25
Cliente:				
DNI: 67213455				
Productos Entregados:				
Codigo	Descripcion	Cantidad	Precio Unitario	Subtotal
NR60310	Pincel No. 2	2	15.25	30.50
...
Productos no entregados				
Codigo	Descripcion	Cantidad		
VG6534	Lapicero Rojo F14	10		
...		
=====				
Pedido No. ...	Fecha:	...	Hora:	...
...				

Consideraciones:

- Los archivos CSV y TXT solo se pueden leer una vez.
- 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 descritas deben realizarse en el orden indicado, no se calificarán las tareas desarrolladas si las previas no se desarrollaron.
- En la calificación se tomará en cuenta el uso correcto de las funciones para leer las cadenas de caracteres, el buen uso de los punteros y de la memoria dinámica asignada, deberá liberar los espacios que ya no se usen, no deberá dejar espacios de memoria sueltos, tampoco deberá gestar memoria dinámica innecesariamente.

ADVERTENCIAS:

- Obligatoriamente debe desarrollar su proyecto bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo.
- Al finalizar el examen, comprima la carpeta **“TP20212_Examen_Final”** 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. Luego, súbalo a la tarea programada en Paideia para este examen.

CRITERIOS DE CALIFICACIÓN:

1. Si el programa presentado tiene más de tres errores de sintaxis serán calificados sobre la mitad del puntaje.
2. Si el programa no muestra los resultados o los que muestran 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 para separar las palabras compuestas (p. e.: baseInf).
4. Se descontará 15% de la nota si no se colocan comentarios relevantes, incluyendo un encabezado al inicio del programa en el que se indique el nombre del autor, la fecha, y una descripción de lo que hace el programa).
5. No se calificará el código puesto como comentario.
6. No se calificarán aquellas funciones implementadas en el archivo main.c

San Miguel, 06 de diciembre del 2021