

TRABAJO PRACTICO INTEGRADOR
PROGRAMACION I
PROF: Mg. Ing. Gastón Addati

Modalidad de presentación: El trabajo es de carácter grupal (2 integrantes por cada grupo).

Para la realización de este trabajo podrán utilizarse diversas tecnologías (herramientas de MS Office, Draw.io, Powerpoint, etc). Eso será decisión de cada grupo.

Principalmente lo que se espera es un documento en formato WORD o PDF y un programa realizado en lenguaje C.

La calificación del trabajo estará basada en: la calidad del trabajo presentado (contenido, forma, prolijidad, entendimiento, etc.). En la cantidad de información recopilada para dar respuesta a las preguntas y al ejercicio. En la solidez para la defensa del trabajo de la sección práctica y en la cantidad de código fuente entregado.

Estructura del Trabajo a presentar:

- Carátula (con logo de UCEMA y datos de los alumnos)
- Índice
- Explicación detallada del problema
- Explicación detallada de la solución, apoyada con diagramas.
- Explicación del funcionamiento del programa
- Conclusiones
- Anexos

El total del trabajo no debe exceder las 40 páginas. Se debe utilizar letra calibri, tamaño 11, interlineado simple. Todo el texto justificado.

Cada página debe tener numeración como pie de página. Nombre de los integrantes del grupo como encabezados.

Descripción del entregable:

Documento PDF o WORD con la estructura mencionada + Código fuente del programa + Ejecutable.

Cronograma de Entregas:

Explicación general para todos los grupos y entrega de los Enunciados: Martes 17/11/20

Revisión y avances grupales. Dudas o consultas: Viernes 20/11

2do parcial: 27/11

Fecha de Entrega TP's: Viernes 4/12.

Defensa del trabajo: Si fuera necesario y Sólo para quien deba recuperar algo o que no llegue a la promoción: 11/12

TEMA 1.

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Una empresa metalúrgica le encarga a usted la construcción de un sistema de ventas, el cual, por una cuestión de funcionamiento, se le solicita que la información sea persistida en archivos tradicionales (archivos de texto).

El Sistema de ventas contiene clientes para los cuales se tiene:

Nombre
Apellido
Direccion
Fecha Ultima compra

El campo Dirección a su vez, está compuesta por

Calle
Altura
CP
Piso
Depto

La empresa vende PRODUCTOS y vende SERVICIOS.

Productos

Nombre de Producto
Descripcion del Producto
Cantidad en stock
Costo (lo que le cuesta a la metalúrgica comprarlo o fabricarlo)

Servicios

Nombre de Servicio
Descripcion del servicio
Costo_hora (el costo por cada hora de servicio)
Cantidad de horas a brindar como servicio
Precio del servicio (calculo automático entre el costo y la cantidad de horas a brindar)

Se pide que construya un programa que permita:

- A. *Ingresar nuevos clientes*
- B. *Ingresar productos*
- C. *Ingresar Servicios*
- D. *Gestionar nueva compra: Esto es, un cliente que quiere comprar un producto o un servicio.*

Para la gestión de la compras:

Se pide que piense en grupo que acciones deberán realizar. ¿Cómo gestionaría las compras de esta empresa? Que cosas harían?.

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 2.

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

UD trabaja en un banco, y se le pide que realice un programa que permita ingresar los siguientes datos de los clientes:

Numero de Cuenta
Datos del Cliente
Ciudad del cliente
Balance de la cuenta

Datos del cliente a su vez es otra estructura que contiene:

Nombre
Apellido
DNI
Edad

Esos datos serán guardados en un archivo binario llamado "clientes.dat".

El programa debe permitir leer del archivo "clientes.dat" los datos que tiene cargados, y mediante un menú interactivo, se deben listar los datos cargados en el archivo "clientes.dat", filtrando por balance (Cero, Negativo, Positivo).

A su vez, el programa debe generar otro archivo binario, llamado **reporte.dat** que permita, utilizando una lista enlazada, tomar la información del archivo clientes.dat y que permita generar y mostrar los clientes ordenados alfabéticamente con el balance de cada uno de ellos.

En un tercer archivo, que se generará cuando el usuario seleccione la opción "reporte", se deberá almacenar en un archivo de texto (**reporte.txt**) todos los clientes que tienen saldo negativo, ordenados alfabéticamente. Este programa, para que funcione correctamente, deberá disponer de un menú de opciones que ud. deberá diseñar sobre la base de lo explicado anteriormente.

Se pide adicionalmente que UD. decida que otra función se puede implementar que pueda ser de utilidad para el banco.

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 3

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Una empresa industrial fabrica y vende tubos de acero, que se utilizan para las perforaciones de pozos de petróleo.

Los productos los comercializa en todo el continente americano, y recibe una gran cantidad de pedidos por hora lo que implica tener equipamiento informático de mucha robustez y potencia para procesar la información. Hay otro sistema de pedidos en funcionamiento, pero por una decisión de la gerencia, se le pide a ud. que utilizando estructuras dinámicas (la que ud. considere) genere un programa que permita tomar la información del sistema principal de pedidos, y almacenar en un archivo los pedidos recibidos en tiempo real. Todo lo que sucede entre el cliente y la generación del pedido, no interesa en esta instancia. Haga de cuenta que la información a ud. le llega en tiempo real.

Los datos que se deben almacenar son:

Cliente (razon social)
Telefono
Ubicación_Cliente
Producto pedido
Cantidad pedida
Precio de venta unitario
Precio de venta total.

Ubicación_Cliente:

Id ubicación
País
ciudad

Producto pedido:

Id Producto
Descripción

El precio de venta total, debe ser calculado automáticamente a partir del precio unitario cargado y de la cantidad pedida.

Por razones de seguridad se pide que los datos ingresados estén alojados en la memoria y que cada 10 productos que se pidan, se vuelquen al archivo pedidos.txt, se libere la memoria una vez que los datos estén grabados en el archivo, y que el proceso vuelva a comenzar.

En un archivo llamado reporte.txt se debe permitir generar, cuando el usuario del programa lo disponga (seleccionando la opción correspondiente) un reporte de la cantidad de dinero vendida por día, indicando cual fué el cliente que más compró (en términos de dinero y no en términos de cantidad de productos), e indicar de que país y ciudad es el cliente que más compró.

Que otra función considera que puede ser implementada en este sistema, que permita otorgarle un valor agregado a la empresa que fabrica los tubos?. Implemente dicha función.

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 4

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Una empresa que vende productos, recibe una solicitud de cualquiera de sus clientes registrados. Al recibir la solicitud, la misma se coloca en una cola de espera (dinámica). Cuando el supervisor lo decide, el producto sale de cola, para colocarse en una pila. Esa pila virtualmente es el stock.

Cuando el producto sale del stock, significa que el cliente lo recibió y que la empresa lo facturó.

Por lo tanto, se pide que haga un programa que permita registrar los pedidos, y sobre todo que permita gestionar el stock (Ingresando pedidos o bien, quitándolos).

Los clientes se deben ingresar manualmente y los mismos deben ser almacenados en un archivo clientes.

Los productos se almacenan en un archivo llamado productos

Cuando un producto sale del stock, el archivo reporte contiene los datos del producto entregado, del cliente que solicitó y del precio de venta.

El archivo de ventas se genera en cualquier momento del programa, y permite ver la facturación total emitida de ese día y hasta ese momento, por la empresa, y debajo, el detalle de cómo se compone dicho número de facturación, es decir, todos los clientes, productos y precios de cada pedido. Se pide que este listado este ordenado de mayor a menor.

En cualquier momento se tiene que poder visualizar los pedidos en cola de espera, y los pedidos en stock.

Datos de los clientes:

Id_Cliente: Es un numero entero, automático y consecutivo que coloca el programa cuando se ingresa un nuevo cliente.

Razon_Social

Dirección

Dirección es un struct anidado con los siguientes campos

Calle

Numero

Localidad

CP

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 5

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Se trata de un sistema que permite gestionar un estacionamiento pago por hora.

Cuando un vehículo ingresa, se le toman los datos, como ser: Marca, Modelo, y Patente. Además, se le debe indicar la hora de ingreso (esto se debe hacer automáticamente por sistema). Si un auto existe, porque fue cargado previamente, no es necesario introducir todos los datos. Se completan con los datos que fueron cargados anteriormente. Por esto motivo, los datos de los vehículos deben estar en un archivo llamado autos.

Los autos ingresados, deben incorporarse a una lista enlazada, con los datos que correspondan. A medida que un nuevo auto ingresa, un nuevo elemento se incorpora en la lista. Cuando un auto se retira, se elimina de la lista, previo al cobro. El garage tiene espacio para 20 autos. Si todos están ocupados, el sistema alertará que no hay lugar, y por lo tanto ese vehículo pasa a esperar en una cola. Cuando se hace espacio, porque un vehículo se fue, el auto de la cola, se inserta automáticamente a la lista.

Cuando un auto se desea retirar, debe buscársele en la lista, y una vez identificado el programa debe calcular cuanto tiempo estuvo en el sistema. Por cada minuto de permanencia, el sistema cobra \$10. Dicho valor debe ser fácil de cambiar o modificar.

El sistema debe permitir obtener reportes tales como:

- a) Cantidad de autos que están estacionados (Elementos de la lista)
- b) Cantidad de dinero cobrado hasta el momento. (Es decir, que cuando un auto se va, se factura el importe que fue calculado en función del tiempo que estuvo).
- c) Buscar autos (es una función que busca un vehículo por patente y muestra si existe o no, y cuánto tiempo lleva estacionado).

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 6

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Se trata de un sistema para gestionar la venta de un local de comidas rápidas, que solo vende dos productos (Salchichas y/o hamburguesas).

Los productos tienen la siguiente estructura

ID_Producto: Es un numero entero y consecutivo que comienza con el 101.

Nombre_Producto:

Precio_Producto

Cantidad:

El pedido que realiza un cliente, se realiza de un único producto por vez (uno a la vez).

Los pedidos se ingresan a una cola de espera, mientras son preparados.

En el menú principal del programa se debe mostrar en todo momento, la cantidad de elementos que están siendo preparados.

Cuando el producto está listo para ser retirado, el operador del sistema, indica que el producto está listo, y el mismo sale de la cola, para incorporarse como información a un archivo donde se registran todos los productos que fueron vendidos y entregados. Indicando para esto, los datos de los producto, con las cantidades y el precio.

El sistema debe tener la posibilidad de emitir reportes de cuantos productos están en preparación. Cuantos productos fueron entregados hasta el momento y al finalizar (cuando el sistema termina o cuando se sale del mismo) se debe crear un archivo log.txt donde se indiquen dos renglones:

- 1) Cantidad de productos vendidos y
- 2) cantidad de dinero facturado.

NOTA: Tendrá un puntaje adicional el hecho de hacer este programa, y que dentro del pedido se puedan incorporar varios productos para el mismo cliente.

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 7

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Se trata de un programa que permita gestionar la entrada a un evento de teatro.

Este teatro es muy particular, porque registra a los clientes antes de vender la entrada al espectáculo. Los datos que registra son: el DNI, el nombre y apellido, además de identificar al mismo como VIP o NO VIP.

Si un cliente es VIP, recibe un 20% de descuento al momento de pagar la entrada.

El teatro tiene una capacidad máxima de 200 personas, por lo que el sistema no podrá vender más de 200 entradas.

Cada vez que una entrada se vende, a un precio fijo, (Este precio fijo debe ser fácil de modificar), se asigna el número de asiento, el cual es un número entero, consecutivo del 1 al 200.

Piense a la cola de clientes como una cola dinámica, y a los asientos del teatro como una estructura pila. Cuando la función del teatro finaliza, los clientes se van en el siguiente orden: desde el último que ingresó al primero que lo hizo (desapilar).

El programa debe ir mostrando progresivamente como se van los clientes del teatro, mostrando en pantalla (con un delay de 2 segundos) los datos de los clientes que se van. EJ:

2002s199.....2s1982s etc.

Piense en un algoritmo que permita establecer cuando el cajero está ocupado. Cuando eso sucede, si justo se quiere llegar otro cliente debe esperar a que pueda ser atendido, es decir, se debe encolar.

El menú de opciones del programa debe permitir entre todas las opciones mencionadas, almacenar en archivos los siguientes datos:

Cantidad de entradas vendidas

Cantidad total de dinero recaudado

Cantidad total de clientes VIP que asistieron a la función

Además en cualquier momento el cajero puede ver cuantas entradas disponibles quedan.

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.

TEMA 8

Explique cómo resolvería el siguiente problema y codifique un programa que brinde una solución. Utilice código, gráficos, tablas, esquemas, todo lo que necesite para explicar el proceso de resolución. Deberá explicarse al máximo nivel de detalle, cada una de las estructuras, funciones, algoritmos y por supuesto, el funcionamiento del programa.

Descripción del problema:

Se trata de hacer un sistema de pedidos de compra de materiales para la construcción. Los pedidos con toda la información asociada a éste, se deben almacenar en uno o mas archivos. Existen varios vendedores simultáneos que usan el sistema, por lo que los pedidos se irán almacenando en memoria y a medida que se entreguen al cliente final, desaparecen de la memoria, pero no así del sistema, el cual debe guardar toda la información asociada, como por ejemplo, el cliente, los productos facturados y el importe total de la factura emitida.

Dado que los pedidos no necesariamente se entregan uno detrás del otro, y en el orden que solicitaron, se pide que implemente un sistema que permita gestionar y llevar un registro de cada transacción.

Deberá utilizar estructuras, archivos, y las estructuras de datos que considere mas oportunos.

Considere lo siguiente:

Los clientes tienen la siguiente información:

ID_cliente: Es el numero asignado a cada cliente. Se carga automáticamente con Números consecutivos desde el cliente 1.

Razón Social:

Dirección: Calle, Numero, Depto, Piso, CP

Zona: ID_Zona, Nombre_Zona (Ej: Zona 1 – La boca, Barracas)

Los productos:

Id_Producto

Descripcion

Stock: Indica cuantos se cargarán en el stock

Precio

El sistema tiene que permitir agregar nuevos clientes, y productos (que además deberán persistir en archivos) y por supuesto, procesar pedidos para cualquier cliente existente. Solo se podrán vender productos cuyo stock sea mayor a 0.

Implemente una función que permita reportar la facturación diaria de la empresa y otra que muestre todos los pedidos ordenados por importe de mayor a menor.

NOTA: Un pedido es único para un cliente. Si el mismo cliente desea comprar otro producto adicional, se debe generar un nuevo pedido.

Tanto para los tipos de datos a utilizar, las estructuras de datos que se elijan implementar, así como cualquier otra información que no esté mencionada o explicitada en este enunciado, deberá definirla ud mismo con su respectiva justificación.

Cuestiones a considerar: Uso de Punteros, Funciones, Archivos, Memoria dinámica, Structs, Pilas, Colas, Listas, etc. Utilice la combinación necesaria para dar la mejor respuesta al problema.

- El código debe estar 100% comentado
- La información en los archivos debe persistir.
- Siempre debe ser posible consultar el contenido de cada archivo desde el programa.