

Universidad Nacional de La Matanza

Temas de Programación 3635 - Comisión 06-4900 (Jueves - Noche)

Examen Parcial - 14/nov/2024

Se ha implementado un sistema de gestión de inventarios en una cadena de tiendas **TiendaMax**. La cadena está formada por un almacén central y 5 tiendas que operan como punto de venta, cada tienda está numerada de 1 a 5. Se requiere un proceso que gestione las operaciones y actualizaciones del inventario basado en el historial de ventas de productos en cada tienda.

A continuación se detalla la información disponible:

productos.bin: Archivo binario con la lista de productos disponibles en todas las tiendas. Cada registro contiene:

- **idProducto**: Número entero que identifica al producto.
- **nombreProducto**: Nombre del producto (máximo 50 caracteres).
- **categoria**: Categoría del producto (máximo 20 caracteres).
- **precio**: Precio del producto.
- **stock**: Cantidad en stock en el almacén central.

ventas_[tienda].csv: Archivo de texto variable separados por comas (,) que registra las ventas de cada tienda. [tienda] representa el número de la tienda (por ejemplo, **ventas_1.csv** para la tienda 1 y **ventas_5.csv** para la tienda 5). Cada registro contiene:

- **idProducto**: Número entero que identifica al producto.
- **cantidad**: Cantidad vendida del producto.

Nota: Puede haber más de una línea de venta para un mismo **idProducto**, y el archivo no está ordenado por ninguno de los campos.

Ejercicio 1

Actualizar el stock en el archivo **productos.bin** a partir de los movimientos registrados en las ventas de todas las tiendas operativas.

Ejercicio 2

Generar el archivo **reposicion.txt** de ancho fijo, el cual contendrá en cada línea **idProducto** y la cantidad necesaria a reponer para que el inventario contenga al menos 10 unidades de cada producto. El archivo generado debe utilizar 5 caracteres para cada campo.

Ejercicio 3

Mostrar por pantalla:

- El producto con más cantidad de unidades vendidas
- La tienda que generó más facturación por ventas
- La categoría más vendida en cada tienda.

Ejercicio 4

En cada ciclo de operaciones, puede que alguna de las tiendas no se encuentre operativa. Indicar por argumento al programa la cantidad de tiendas operativas (un numero entero de 1 a 5 inclusive, que debe ser validado). Solo se tomarán en cuenta las tiendas operativas según lo indicado en el argumento.

Ejemplos

productos.bin (Lista de productos)

idProducto	nombreProducto	categoría	precio	stock
2	Leche	Lacteos	0.99	30
5	Yogur	Lacteos	1.10	10
1	Manzanas	Frutas	1.25	50
4	Jugo de Naranja	Bebidas	1.50	25
3	Pan	Panaderia	2.50	20

ventas_1.csv (Ventas de la tienda 1)

1, 10
2, 5
3, 15
5, 2

ventas_2.csv (Ventas de la tienda 2)

1, 5
3, 10
4, 8
5, 3

ventas_3.csv (Ventas de la tienda 3)

1, 8
2, 10
3, 7
4, 5

ventas_4.csv (Ventas de la tienda 4)

2, 8
3, 6
4, 10
5, 4

ventas_5.csv (Ventas de la tienda 5)

1, 7
2, 12
3, 5
4, 7

productos.bin (Luego de la actualización de las 5 tiendas operativas)

idProducto	nombreProducto	categoría	precio	stock
2	Leche	Lacteos	0.99	15
5	Yogur	Lacteos	1.10	1

idProducto	nombreProducto	categoría	precio	stock
1	Manzanas	Frutas	1.25	20
4	Jugo de Naranja	Bebidas	1.50	-5
3	Pan	Panaderia	2.50	-18

reposicion.txt (Productos a reponer)

```
1    0
2    0
3   28
4   15
5    9
```

Salidas

Producto con más cantidad de unidades vendidas:
Pan: 43 unidades vendidas.

Tienda que más facturación generó por ventas:
Tienda 2: \$34.75

Categoria mas vendida en cada tienda:

Tienda 1: Panaderia
Tienda 2: Panaderia
Tienda 3: Lacteos
Tienda 4: Bebidas
Tienda 5: Lacteos

Condiciones de aprobación del examen

- El programa debe compilar correctamente en todos los casos.
- La compilación debe estar exenta de **warnings** arrojados por el compilador.
- Vectores y cadenas de texto deberán ser manipulados utilizando aritmética de punteros.
- Las soluciones tienen que ser eficientes:
 - En el uso de memoria, por tanto, no declare vectores o matrices auxiliares si no es necesario.
 - En cantidad de ciclos de procesador y en el caso de matrices las soluciones deben ser óptimas.
 - Los recorridos de archivos tienen que ser mínimos.
- Nunca acceda a memoria que no le pertenece.
- Declare variables al inicio del bloque y no utilice VLA (Variable length arrays)
- Utilice nombres descriptivos, separe en funciones cuando sea conveniente.
- Las soluciones tienen que estar desarrolladas en ANSI C para garantizar tanto como sea posible compatibilidad multiplataforma.
- Las funciones de manipulación de archivos de texto y funciones de operación de archivos binarios deben ser utilizadas únicamente en los archivos con el formato correcto.
- La interpretación de los requisitos y la estructura de la aplicación es parte de la evaluación.
- Trabajar de forma genérica siempre que sea posible.
- Reducir al mínimo posible la cantidad de lecturas de cada archivo.
- Para superar la marca de “aprobado” se debe desarrollar las funciones solicitadas.
- Para superar la marca de “promoción”, se deben utilizar TDA correspondientes (Vector, Iterador, Indice, Matriz, etc.) mas acorde a la solución. Cada archivo de ventas solo puede ser leído 1 vez.

Instrucciones de Entrega

Agregue sus datos en la cabecera de cada archivo fuente entregado (Apellido, Nombre y DNI). De comentar los **define** correspondientes a los ejercicios entregados.

```
// Apellido, Nombres:  
// DNI:  
//  
// Entrega:  
// #define ENTREGA_EJERCICIO_1_SI  
// #define ENTREGA_EJERCICIO_2_SI  
// #define ENTREGA_EJERCICIO_3_SI  
// #define ENTREGA_EJERCICIO_4_SI
```

Comprima los fuentes de la solución en un archivo `.zip`, el formato de dicho archivo debe ser `PAR_DNI_Apellido_Nombre.zip`.