

TÉCNICAS DE PROGRAMACIÓN
CUARTO LABORATORIO CALIFICADO
SEMESTRE ACADÉMICO 2021-1

Horario: Todos

Duración: 110 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 LABORATORIO SERÁ CONSIDERADO UNA FALTA DE PROBIIDAD 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.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS 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, DE LO CONTRARIO SE LE DESCOTARÁ 0.5 PUNTOS POR CADA OMISIÓN. NO SE HARÁN EXCEPCIONES.

Cree un proyecto en NetBeans con el nombre: **“Laboratorio04-2021-1”**(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.

Un restaurante recibe cada mes un archivo de texto de uno de sus proveedores, la lista de precios de los productos que venden, el archivo es similar al siguiente:

```
ListaDePrecios.txt
Bodega 2000 S. A.
32115 [Arroz blanco] 3.56
22345 [Papa blanca] 1.62
...
```

La primera línea del archivo contiene el nombre de la compañía proveedora, a partir de allí vienen los datos de cada producto, esto es: código, nombre y precio del producto por unidad.

Por otro lado, el restaurante posee un archivo de texto en el que se almacenan todos los platos que podría preparar. El archivo tiene la forma siguiente:

```
Platos.txt
12345 (Arroz con pollo) 35407 1.2535407 6.7 28775 1.23 33567 0.82 ...
11345 (Papa a la huancaína) 14032 35407 6.7...
...
```

Primero viene el código del plato, le sigue el nombre del plato y por último vienen todos los ingredientes (productos) que requiere para preparar el plato (código del ingrediente y cantidad requerida).

El restaurante cada cierto tiempo revisa sus ventas y de acuerdo con ellas modifica la carta de platos que ofrece a su clientela, eliminando o reduciendo los platos poco vendidos y agregando o aumentando platos que cree pueden ser más atractivos. Esta información la guarda en otro archivo de texto, como se muestra a continuación:

```
platosAPreparar.txt
1234556
11345162
...
```

En él se almacena en cada línea los datos de los platos que se elaborarán diariamente (código del plato y cantidad de platos a preparar).

En el restaurante, ocurrió un desperfecto y **todos los productos almacenados para preparar los respectivos platos se descompusieron**. El archivo *productosAlmacenados.txt* contiene la siguiente información:

productosAlmacenados.txt	
32115	12.35
22345	10.45

En cada línea se muestra para un producto: el código y la cantidad de productos almacenados.

El restaurante, requiere un reporte para controlar el costo de las pérdidas en las que incurrirán por la descomposición de los productos y la suma de los precios de los ingredientes de los platos que no podrán vender porque no se podrán preparar.

Reporte de Peridas.txt

PRODUCTOS DESCOMPUESTOS				
Producto	Cantidad	Precio Unitario(S/.)	Sub-total(S/.)	
1) 07899	9.65	12.1	116.75	
...				
20) 32115	2.09	3.9	6.27	
...				
Costo total de productos descompuestos: S/. 12389.67				
=====				
PLATOS QUE NO SE PODRÁN PREPARAR				
Plato	Cantidad	Precio Ingredientes(S/.)	Sub-total(S/.)	
1) 12345	20	15.35	307.00	
...				
5) 11345	15	18.22	273.30	
...				
Suma de precios de los ingredientes de los platos que no se podrán preparar: S/. 18654.00				
=====				

El reporte está organizado en dos partes:

- En la primera parte, debe mostrar por cada producto descompuesto, la cantidad perdida, el precio unitario del producto y el sub-total (cantidad x precio unitario). Debe mostrar también el costo total de los productos descompuestos (suma de sub-total por producto). En el reporte no deben mostrarse los productos que no se encuentran en el almacén. **La información de los productos se muestra ordenada en forma descendente por el sub-total.**
- En la segunda parte, debe mostrar por cada plato, la cantidad de platos que no se podrán preparar, la suma del precio de los ingredientes que se utilizan para cocinar cada plato y el sub-total (cantidad x precio de los ingredientes). Debe mostrar también la suma de los precios de los ingredientes de los platos que no se podrán preparar (suma de sub-total por plato). **La información de los platos se muestra ordenada en forma ascendente por la cantidad y descendente por el sub-total.**

Para desarrollar el reporte, los archivos *ListaDePrecios.txt*, *productosAlmacenados.txt*, *platos.txt* y *platosAPreparar.txt* solo se pueden leer una sola vez. Coloque en arreglos los datos de estos archivos que considere convenientes, como máximo se podrán tener 300 productos y 100 platos.

Debe asumir que en el almacén siempre habrá cantidad suficiente de todos los productos para preparar los platos, por lo tanto, ninguno de los platos se podrá preparar. Si el producto no se encuentra en el almacén no se debe mostrar en la primera parte del reporte. Si existe un producto en el almacén que no tiene precio registrado no se debe colocar en la primera parte del reporte. Si el plato a preparar no tiene registrado los ingredientes, no se debe considerar en la segunda parte del reporte. Si algún ingrediente de los platos no se encuentra en el archivo *ListaDePrecios.txt*, debe asumir que aporta S/.0.0 al precio de los productos del plato.

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.

En esta pregunta se tomará muy en cuenta en la nota: el manejo correcto de los arreglos y la información que almacenan; la correcta lectura de los datos; en este sentido será obligatorio leer los números como tal, no permitiéndose su lectura carácter por carácter o como cadenas, y el formato del reporte; éste deberá ser lo más parecido a la muestra dada. No se podrá emplear el carácter de tabulación ('\t') para la emisión del reporte. No puede usar el tipo de dato cadena de caracteres para leer los datos.

ADVERTENCIAS:

- Obligatoriamente debe desarrollar su proyecto bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo.
- **Al finalizar el laboratorio, comprima la carpeta de su proyecto en un solo archivo de tipo .zip** (Solo debe usar el programa Zip que viene por defecto en el Windows, **NO** deberá usar Win-RAR, RAR, 7Zip, etc.) y súbalo a la tarea programada en Paideia para este laboratorio.

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 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 programa).
5. No se calificará el código puesto como comentario.
6. No se calificarán aquellas funciones implementadas en el archivo main.c

San Miguel, 01 de junio del 2021