

Examen final – 12 de septiembre de 2011

Tiempo disponible: 2 horas

Ejercicio único

Se pide construir un programa (**septiembre.cpp**) que gestione una lista de recetas. Cada receta consiste en un número de receta, un nombre, una lista de ingredientes y el total de calorías de la receta. Cada ingrediente consiste en un nombre, una cantidad en gramos y las calorías por gramo de ese ingrediente.

Programa principal (1 punto)

Comenzará creando la lista con la información de los archivos. Si algún archivo no existe, lo indicará y terminará la ejecución. Si la carga tiene éxito, ordenará la lista como se indica más adelante y ofrecerá a continuación un menú al usuario:

- 1 - Mostrar la lista de recetas
- 2 - Receta baja en calorías
- 0 - Salir

El menú se repetirá hasta que el usuario elija la opción 0.

Estructuras de datos (1,5 puntos)

Debes declarar las estructuras de datos adecuadas para la información que se va a obtener de los archivos de entrada (*apartado siguiente*).

Carga de la lista de recetas (2,5 puntos)

Se parte de un par de archivos de texto con los datos de las recetas (**recetas.txt**) y de los ingredientes de las recetas (**ingredientes.txt**). Cada uno contiene una serie de líneas, cada una con la información de una receta o ingrediente y estando los datos separados por un espacio:

- ✓ Cada línea de **recetas.txt**: número de receta (entero sin signo) y nombre de la receta (cadena de 40 caracteres como máximo). Máximo 500 recetas.
- ✓ Cada línea de **ingredientes.txt**: nº de receta (entero sin signo), gramos (real), calorías por gramo (real) y nombre del ingrediente (cadena de 40 caracteres como máximo). Máximo 10 ingredientes/receta.

Ambos archivos terminan con una línea en la que sólo hay un 0 indicando su fin.

Ejemplos de archivos de entrada:

<code>recetas.txt</code>	<code>ingredientes.txt</code>
123 Ensalada_de_arroz 2431 Pechugas_de_pollo_con_almendras 37 Ensalada_de_pollo 0	123 100 3.4 Arroz 123 250 0.13 Lechuga 123 150 0.22 Tomate 123 60 8.6 Aceite 123 15 0.6 Salsa_de_soja 2431 250 1.7 Pechugas_de_pollo ... 37 125 5.8 Salsa_rosa 0

Se creará la lista de recetas a partir de la información de los archivos. La información en los archivos no tendrá errores (nº de receta inexistente, valores de tipos incorrectos o exceso de ingredientes o recetas), pero hay que comprobar que se puedan abrir. Además, las cadenas no contienen espacios en blanco.

Ordenación de la lista de recetas (1,5 puntos)

Ordenará la lista de mayor a menor nombre de receta. La ordenación de la lista se realizará por el método de selección directa y se implementará de forma recursiva.

El subprograma de ordenación recibirá la lista y la posición del array que se procesa en cada llamada (0, 1, 2, ...):

```
void ordenar(tLista &lista, int pos);
```

El subprograma localizará el registro con mayor nombre de lo que quede de lista (desde `pos`) y lo intercambiará con el de esa posición `pos`, para a continuación realizar una llamada recursiva para que se ordene lo que resta de lista.

1 - Mostrar la lista de recetas (1,5 punto)

Mostrará la lista de recetas con el siguiente formato de ejemplo para cada una:

```
123 Ensalada_de_arroz (930.5 calorías): 100 gr. de Arroz, 250 gr. de  
Lechuga, 150 gr. de Tomate, 60 gr. de Aceite, 15 gr. de Salsa_de_soja.
```

Las calorías, con exactamente un decimal.

2 - Localizar recetas bajas en calorías (2 puntos)

Pedirá un máximo de calorías y localizará la primera receta con menos de ese máximo (si existe). Lo devolverá a través de un puntero a una variable dinámica (que se creará antes de llamar al subprograma de búsqueda). Si se ha encontrado, mostrará su información con el mismo formato del apartado anterior.

Cuida el estilo, define todos los tipos necesarios y estructura bien el código.

Una vez que hayas terminado el ejercicio, *súbelo* al Campus Virtual:

Tarea identificada como **Examen de septiembre: Ejercicio**

¡No olvides poner al principio del programa un comentario con tu nombre!