

## Algoritmos y Programación I (95.11) – Curso Essay – 3<sup>er</sup> parcialito – 05/07/2021

Resolver los siguientes problemas en un único archivo de código ISO-C99.

1. Se quiere modelar un TDA que sirva para que un restaurant lleve el pedido de cada una de las mesas. Los comensales durante su estadía van pidiendo diferentes ítems que se van consignando en el pedido. Se pide:
  - a. Declarar la estructura que encapsula el TDA `pedido_t` que representa el pedido de una mesa particular. Explicar qué representa cada miembro y documentar el invariante de representación.
  - b. Implementar la primitiva `size_t pedido_cantidad(const pedido_t *p, const char *item);` que dado el nombre de un `item` indique cuántas unidades del mismo se pidieron en el pedido `p`.
  - c. Implementar la primitiva `bool pedido_agregar(pedido_t *p, const char *item);` que agregue un `item` al pedido `p`. La función debe devolver `true` de poder realizar la operación.

2. a. Implementar la función

```
bool leer_linea_csv(FILE *f, char columnas[] [MAX_STRING], size_t *n)
```

que dado un archivo de texto `f` en formato CSV lea de él una línea y guarde cada una de las `columnas` de la misma. Se debe devolver a través de `n` la cantidad de columnas leídas y por el nombre `true` en caso de haber sido correcta la lectura. Asumir que tanto la cantidad de `columnas` como `MAX_STRING` son suficientes para realizar la operación.

- b. Proponer un **formato binario** que sea “cómodo” para volcar una fila de un archivo CSV.

Implementar la función

```
bool escribir_fila(const char *ruta, const char *columnas[], size_t n);
```

que dado el nombre del archivo **binario** `ruta` agregue en él las `n` `columnas` conforme al formato propuesto. La función debe devolver `true` en caso de poder realizar la operación.

Ayuda: Para agregar algo al final de un archivo puede ser abierto en modo “a”, *append*.

Nota: Cuando se diseñe el formato pensarlo para que sea no sólo fácil de escribir sino fácil de leer después.

3. Escribir un programa que se ejecute como:

```
$ ./cut [archivo] [c0] <[c1] <[c2] ...>>
```

que dado un **archivo** de **texto** en formato CSV muestre por `stdout` sólo las columnas `cn` indicadas.

Por ejemplo si se tiene el `archivo.csv`:

```
Tomates,2.00,200
Pollo,2.5,600
Papa,5.0,210
```

y se invoca

```
$ ./cut archivo.csv 1 2
```

la salida deberá ser:

```
2.00,200
2.5,600
5.0,210
```

Si bien cada uno de los ejercicios es independiente de los demás se deberán codificar todas las funciones en un único archivo de código fuente que será el que se entregue.

Siendo que el ejercicio 3. es un programa, si se necesitara probar los otros ejercicios hacer una función para tal fin.

El examen es de elaboración personal, todo el código entregado debe ser realizado por el alumno.

El examen se envía a través del sistema de entregas de trabajos prácticos.

¡Suerte! :)