

Laboratorio 8

Programación 1

Segundo semestre 2017

1. Escribir una función que recibe como parámetros:

- a) el nombre (ubicación y nombre con extensión) de un archivo
- b) una cadena de caracteres

y retorna la cantidad total de veces que esa cadena aparece en el archivo.

(Nota: La cadena puede estar formada por uno o varios caracteres. Por ejemplo, un espacio " ", una letra "z", o por varios caracteres "¡Hola!").

2. Escribir una función que abra 2 archivos indicados por parámetro y escriba un nuevo archivo con las líneas intercaladas de los archivos originales.

3. Escribir una función que le permita ingresar texto a un usuario, hasta que ingrese la secuencia de caracteres "::~q" con la cual indicará que finaliza el ingreso.

Luego de finalizado el ingreso, el programa le solicitará que ingrese un nombre (ubicación y nombre con extensión) de archivo en el cual se grabará todo el texto ingresado.

Finalmente le preguntará si quiere ver el texto guardado en el archivo, y en caso afirmativo le mostrará el contenido del archivo grabado.

4. Escribir una función para buscar las ubicaciones en términos de línea y posición, de una cadena que se recibe como parámetro.

La función deberá tener los siguientes parámetros:

1. el nombre de un archivo (ubicación)
2. una cadena con los caracteres a ser buscados en el archivo
3. una cadena con el nombre y apellido de quien realiza la consulta

La función deberá realizar la búsqueda y generar la lista cuyas características se detallan a continuación:

a) En la primera posición de la lista deberá cargarse la cantidad total de veces que aparece la cadena en el archivo

b) A partir de la posición 1, los sucesivos elementos de la lista serán "listas de dos elementos", que tendrán

b.1) En la primera posición ==> un número de una línea en la que se encontró la cadena,

y

b.2) En la segunda posición ==> la ubicación; dentro de la línea; a partir de la cual se encuentra la cadena, empezando a contar desde la posición 0.
Nota: Una cadena puede aparecer más de una vez en una misma línea y la lista a ser retornada deberá contener la información de ubicación de todos los casos.

Antes de retornar la lista, la función le preguntará al usuario si quiere ver el resultado de la búsqueda, y en caso afirmativo le mostrará la información contenida en la lista, con el agregado de texto que explique el resultado, según el siguiente formato que se deberá cumplir tal como se detalla a continuación:

```
Línea 1: Estimad@ X...X
(Nota: "X...X" corresponde al contenido del tercer parámetro)
Línea 2: La cadena Y...Y apareció un total de Z..Z veces en el archivo"
(Nota: Y...Y corresponde al contenido del segundo parámetro y Z..Z corresponde a la cantidad total de veces que se encontró la
cadena)
Línea 3: en las posiciones que se detallan a continuación
Línea 4: Nro. de línea          Ubicación
Línea 5: =====
Línea 6:
Línea 7:
Línea 8:
...
Línea X: =====
```

Nota: En caso de que la cadena no aparezca en el archivo, se deberá mostrar el resultado con el siguiente formato:

```
Línea 1: Estimad@ X...X
(Nota: "X...X" corresponde al contenido del tercer parámetro)
Línea 2: La cadena Y...Y no se encuentra en el archivo
(Nota: Y...Y corresponde al contenido del segundo parámetro)
```

5. Escribir una función que permite concatenar archivos en uno sólo, uno a continuación del otro.
La función debe recibir dos parámetros:
 1. una lista de archivos
 2. un nombre de archivo de salida (ubicación y nombre)

y crear un nuevo archivo con el nombre indicado en el segundo parámetro, cuyo contenido está formado por los sucesivos contenidos; uno a continuación del otro; en el orden en que aparecen en la lista de archivos recibida como primer parámetro.

6. Escribir una función que busca una cadena en un archivo, y retorna los números de línea en las que aparece la cadena comenzando en 1.
7. A partir de un archivo de texto de notas de la materia Programación 1, donde cada línea tiene el formato:

cédula,nombre,nota final

(Cada línea corresponde a un alumno de la clase (no se repiten) y la nota es un valor numérico del 0 al 100)

- a) Crear una clase "Nota" que permita almacenar los datos de cada alumno y su nota, incluyendo métodos para obtener y modificar valores (getters y setters).
- b) Crear una clase "Grupo", que tenga un nombre de grupo, un salón y una lista de notas.
- c) Crear un método de clase en la clase Grupo, que cargue las notas desde un archivo de notas, recibido por parámetro.

- d) Crear un método en la clase Grupo, que grabe un archivo de texto de nombre ResultadosP1, con solamente los nombres y notas de los alumnos que aprobaron la materia (los que tienen nota mayor o igual a 61), ordenados en forma decreciente por nota.

Desafío

8. Escribir un programa que carga información sobre los clientes de una empresa desde un archivo en formato CSV (valores separados por comas) y permita procesar la información y guardarla en un archivo nuevo.

Ejemplo del archivo original:

RUT,razón-social,dirección,teléfono 1,teléfono 2

El usuario ingresará desde el teclado las opciones de procesamiento que se realizarán con la información (siempre con un “-“ antes del comando):

“-s=X”: los datos serán separados utilizando el separador X (por ejemplo, “-s=;” separará los datos utilizando un punto y coma).

“-c=X,Y,Z” : se incluirán solo las columnas X, Y y Z en el archivo de salida. Las demás columnas no serán incluidas. X, Y y Z son los índices de las columnas, comenzando en 0.

“-a=DDD” : agrega una nueva columna al final con el valor DDD para todas la líneas.

“-o=NNN” : el archivo de salida se llamará NNN.

Todas las opciones de configuración son opcionales, y se separan utilizando un espacio.

Por ejemplo, si el usuario ingresa:

“-o=salida.txt” => toda la información se escribe en salida.txt

“-c=0,1 -o=salida.txt” => las dos primeras columnas de la información de cada cliente se escriben en salida.txt

“-s=# -c=3,4” => Las información de las columnas 3 y 4 de cada cliente se escriben en un archivo de salida por defecto, separadas por “#”.