

Tarea # 1 (6%)

Objetivos Generales

Familiarizarse con los aspectos básicos de programación en lenguaje C.

Objetivos Específicos

- Adquirir destrezas en el lenguaje C
- Adquirir destrezas en la definición de estructuras compuestas y tipos.
- Adquirir destrezas en el uso y manejo eficiente de memoria dinámica.
- Utilizar las herramientas libres básicas de desarrollo de programas en ANSI C (editor de texto, make, compilador GNU, depurador)

Definición del problema

Desarrollar un programa que tome una lista de pares de cadenas de caracteres $\langle cad1, cad2 \rangle$ y reemplace, en un conjunto de archivos de *texto*, todas las ocurrencias de *cad1* por *cad2*.

La sintaxis del comando es:

```
# sustituir <archivo pares> {lista de nombres de archivos}
```

donde

- \langle archivo pares \rangle : Cada línea del archivo tendrá dos cadenas de caracteres separadas por el carácter : (dos puntos).
- {lista de nombres de archivos}: lista de archivos de texto.

El programa deberá utilizar una estructura de lista doblemente enlazada para almacenar los pares de cadena de caracteres, esta lista debe ser definida en base a memoria dinámica.

Ejemplo

- Dados los siguientes contenidos de archivos:

palabras.txt	Archivo1.txt	Archivo2.txt
Hola:hola nada:completo Mer:Rosa tyler:arepera Marcos:Luis Ana:Luisa Analisis:resumen	Hola que tal? No tengo nada que reportar, Salvo que Tyler no se presentó a la reunión. Por lo tanto no tengo el Analisis. Saludos	Hola Marcos, No he podido ir al Mercado pero mandaré a Ana.

Note que:

- Su implementación debe ser *case sensitive*.

- Entre cada archivo debe agregar una línea con los caracteres – (doble guion).
- Asuma que cada cadena *cad1* es única.
- En caso de ambigüedad al momento de aplicar una sustitución, debe darle preferencia a la cadena *cad1* más larga. Vea el caso de las cadenas Ana y Analisis en el ejemplo anterior. En ese caso, las últimas dos líneas podrían ser usadas. Sin embargo, deberá usarse Analisis por ser más larga que Ana.

Para la ejecución siguiente:

```
# sustituir palabras.txt Archivo1.txt Archivo2.txt
```

la salida sería:

```
hola que tal?
No tengo completo que reportar, Salvo que Tyler
no se presentó a la reunión. Por lo tanto no tengo el
resumen.
Saludos
--
Hola Luis,

No he podido ir al Rosacado pero mandaré a Luisa.
```

No realice otros cambios en el contenido de los archivos. La verificación de la ejecución será realizada en forma binaria y automática, es decir, usando un programa que realice la comparación de igualdad entre lo reportado por su aplicación y lo esperado.

Recomendaciones

1. Diseñe su solución completa antes de proceder a implementar.
2. Trabaje en forma ordenada e incremental
3. Pruebe que cada una de sus funciones opera correctamente
4. Estructure bien su código.
5. El código debe tener una cantidad adecuada de comentarios.
6. Realice la documentación de su código a medida que vaya programando, dejarlo para el final se traduce en invertir más tiempo para hacerlo.

Entrega de la tarea:

Realización: Individual

Digital: Hasta las 11:59pm del Jueves semana 4.

Cada estudiante debe colocar en un archivo que con nombre carnet.tar.gz:

- los fuentes de su tarea (archivos .c y .h). Estos deben estar bien identificados, documentados y estructurados. Este conjunto será usado para compilar y correr su tarea.
- Un archivo de nombre *codigo.pdf* con los fuentes de su tarea. Exactamente la misma versión que los .c y .h. Éste será usado para evaluar el código de su tarea.
- El Makefile que pueda compilar/enlazar su aplicación

Note que debe estar suscrito al espacio canvas para poder optar a esta opción, **no espere al día de la entrega para notificar que tiene problemas o que no se ha registrado.**

NOTAS IMPORTANTES:

1. Tarea que no cumpla con algunas de las especificaciones establecidas en este enunciado corre el riesgo de no ser corregida.

Material útil (leer las páginas del manual):

1. Algunas funciones de la librería strings.h
 - strlen, strcat, strcpy, strcmp, strcasecmp
2. Algunas funciones de la librería stdio.h
 - fopen
 - fscanf, scanf, sscanf
 - printf, fprintf, sprintf
3. Algunas funciones de la librería stdlib.h
 - malloc, free
4. <https://overiq.com/c-programming-101/fscanf-function-in-c/>
5. <https://fresh2refresh.com/c-programming/c-printf-and-scanf/>