



Universidad Andrés Bello

Facultad de Ingeniería

Ingeniería en Computación e Informática

FUNDAMENTOS DE PROGRAMACIÓN CONTROL VI

Profesor: Carlos Contreras

Fecha: 8 de Octubre de 2014

Ayudantes: Omar Opazo – Tamara Saéz – Felipe Reyes – Nicolás Venegas

Instrucciones:

- Debe enviar sus resultados en archivos distintos y comprimidos, con su nombre y apellido, al correo electrónico de su profesor de laboratorio.
- Tiene hasta el final de la clase para entregar el control. En caso de no entregar a tiempo, la nota correspondiente es la mínima (1,0).
- El Control es Individual.

Ejercicio 1

Se solicita realizar un programa en C, que cumpla con el siguiente flujo:

Solicitar a un usuario el ingreso de un carácter, denominaremos dicho carácter como COMPARADOR.

Solicitar al usuario el ingreso de n palabras, de largo máximo 30 caracteres. El ingreso de estas palabras finalizará cuando el usuario ingrese la palabra "salir" (sin las comillas). Este será el único modo de dar término al ingreso de palabras.

Posteriormente, las n palabras ingresadas deberán ser concatenadas una tras otra y almacenadas en algún lugar para su posterior uso.

La idea principal es generar un valor denominado como CAMBIADOR (un carácter) a partir de la suma de los subíndices de la palabra concatenada que cumplan con ser igual a COMPARADOR.

Finalmente CAMBIADOR debe superponer a todo COMPARADOR en la palabra concatenada. Existen algunas reglas para el programa:

1. La suma de los subíndices que generará CAMBIADOR, puede ser, a lo sumo, 126.
2. La suma de los subíndices que generará CAMBIADOR, puede ser, a lo menos, 33.
3. La salida del programa, debe enviarse a un archivo "salida.txt".

Ejemplo de entrada:

Ingrese el comparador:

a

Ingrese palabra, para salir escriba 'salir'
hola

Ingrese palabra, para salir escriba 'salir'
chao

Ingrese palabra, para salir escriba 'salir'
salir

Ejemplo de salida.txt:

```
CADENA CONCATENADA holachao
COMPARADOR a
CAMBIADOR *
CADENA MODIFICADA hol*ch*o
```

Debe tener cuidado con el formato de salida, debe ser en el orden especificado y respetando el formato de mayúsculas y minúsculas. Debe utilizar funciones pertenecientes a la biblioteca **string.h**