



PROCESADORES DE LENGUAJE

Ingeniería Informática
Especialidad de Computación
Tercer curso
Segundo cuatrimestre



Departamento de Informática y Análisis Numérico
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico 2017 - 2018

Hoja de ejercicios de FLEX

1. Sumar parametrizado

- Codifica un programa en flex que copie el archivo de entrada en uno de salida sumando N a todo número positivo que sea múltiplo de M, donde N y M son dos números pasados como argumentos desde la línea de órdenes.
- Ejemplo:

```
./sumar_parametrizado.exe Entrada.txt Salida.txt 5 4
```

Entrada.txt

```
1    2    3    4  
5    6    7    8  
9    10   11   12
```

Salida.txt

```
1    2    3    9  
5    6    7    13  
9    10   11   17
```

- Observación:
 - Se recomienda usar la función `atoi()` de C que transforma una cadena de caracteres en su valor entero.

2. Sustitución de una palabra

- Codifica un analizador que reemplace una palabra por otra en un fichero de entrada.
- Ambas palabras, así como el nombre del fichero deberán ser introducidos por el usuario a través de la línea de comandos.
- Al final, el programa deberá indicar cuántas palabras han sido sustituidas.

- Ejemplo:
./cambiar.exe fichero.txt antigua nueva

3. Cifrado César

- El cifrado César o cifrado por desplazamiento es un sencillo método para encriptar información:
 - Cada letra se sustituye por la que se encuentra N posiciones más adelante en el alfabeto.
 - Por ejemplo, si N=3 entonces la conversión es la siguiente
 - Original: ABCDEFGHIJKLMNOPQRSTUVWXYZ
 - Cifrado: DEFQHIJKLMOÑPQRSTUVWXYZABC
- Codifica un programa que permite aplicar el cifrado César.
- El programa podrá ser usado de dos maneras
 - Interactiva:
 - Solamente recibe como argumento la clave de conversión
 - El programa termina cuando se teclea el carácter de fin de fichero
 - `cifradoCesar 3`
 - ✓ `Hola`
 - `Krñd`
 - ✓ `estudiante`
 - `hwxxgldpwh`
 - `cifradoCesar 24`
 - ✓ `Krñd`
 - `Hola`
 - ✓ `hwxxgldpwh`
 - `estudiante`
 - Usando ficheros y la clave de conversión
 - `cifradoCesar 3 entrada.txt salida.txt`
- Observaciones
 - Se deben tener en cuenta las letras mayúsculas y minúsculas.
 - Se debe tener en cuenta el uso de la letra “ñ”.
 - Se valorará el uso de las letras acentuadas.
 - Los caracteres que no sean alfabéticos no se modificarán.
- Referencias
 - El cifrado por sustitución. El Código César.
 - Matemáticas cercanas. Rompiendo los muros de las matemáticas
 - <https://matematicascercanas.com/2017/11/26/cifrado-sustitucion-codigo-cesar/>
 - Cifrado César
 - Wikipedia
 - https://es.wikipedia.org/wiki/Cifrado_C%C3%A9sar

4. Fichero de registro de accesos

- Considérese un fichero de registro que contiene información sobre los accesos de los usuarios a un sistema informático.
- Cada línea tiene la siguiente información
 - usuario fecha hora
- Observación
 - Cada estudiante debe elegir el formato de los usuarios, la fecha y la hora.
- Codifica un analizador léxico que permita
 - Contar cuántos accesos ha realizado un usuario
 - El programa recibirá como argumentos
 - ✓ el nombre del fichero de registro
 - ✓ el nombre del usuario
 - Ejemplo
contar registro.txt i32lusaf@uco.es
 - Contar cuántos trabajos accesos ha realizado un usuario en un día concreto
 - el nombre del fichero de registro
 - el nombre del usuario
 - el día que se desea consultar.
 - Ejemplo
contar registro.txt i32lusaf@uco.es 15-3-2018

5. Analizador léxico de pseudocódigo

- Codifica un analizador léxico que permita reconocer los componentes léxicos de un programa escrito en pseudocódigo.
- Palabras reservadas
 - *inicio, fin, mod, leer, escribir, si, entonces, si_no, fin_si, mientras, hacer, fin_mientras, repetir, hasta_que, para, desde, hasta, paso, fin_para*
 - No se distinguirá entre mayúsculas ni minúsculas.
 - Las palabras reservadas no se podrán utilizar como identificadores.
- Identificador
 - Características
 - Estarán compuestos por una serie de letras, dígitos y el subrayado;
 - Deben comenzar por una letra,
 - No podrán acabar con el símbolo de subrayado, ni tener dos subrayados consecutivos.
 - No se distinguirá entre mayúsculas ni minúsculas.
 - Ejemplos
 - Identificadores válidos:
dato, dato_1, dato_1_a

- Identificadores **no** válidos:
dato, dato, dato__1
- **Número**
 - Se utilizarán números enteros, reales de punto fijo y reales con notación científica.
 - Todos ellos serán tratados conjuntamente como números.
- **Cadena**
 - Estará compuesta por una serie de caracteres delimitados por comillas simples:
'Ejemplo de cadena'
 - Deberá permitir la inclusión de la comilla simple utilizando la barra (\):
'Ejemplo de cadena con \' comillas\' simples'.
 - **Nota:**
 - Las comillas exteriores no formarán parte de la cadena.
- **Operador de asignación**
 - ASIGNACIÓN: :=
- **Operadores aritméticos:**
 - SUMA: +
 - RESTA: -
 - PRODUCTO: *
 - DIVISIÓN: /
 - MÓDULO: mod
 - POTENCIA: **
- **Operador alfanumérico:**
 - CONCATENACIÓN: ||
- **Operadores relacionales de números y cadenas:**
 - MENOR_QUE: <
 - MENOR_IGUAL_QUE: <=
 - MAYOR_QUE: >
 - MAYOR_IGUAL_QUE: >=
 - IGUAL: ==
 - DISTINTO: <>
 - Por ejemplo:
 - Si *A* es una variable numérica y *control* una variable alfanumérica, se pueden generar las siguientes expresiones relacionales:
(A >= 0)
(control <> 'stop')

- **Operadores lógicos:**
 - DISYUNCIÓN_LÓGICA: .o.
 - CONJUNCIÓN_LÓGICA: .y.
 - NEGACIÓN_LÓGICA: .no.
 - Por ejemplo:
(A >= 0) .y. .no. (control <> 'stop')

- **Comentarios**
 - De varias líneas: delimitados por (* y *)

(ejemplo maravilloso
de comentario
de tres líneas *)*

 - De una línea:
 - Todo lo que siga al carácter # hasta el final de la línea.

ejemplo espectacular de comentario de una línea

- **Otro componentes léxicos**
 - FIN_SENTENCIA: ;
 - Paréntesis
 - Izquierdo: (
 - Derecho:)

- **Control de errores**
 - El intérprete deberá controlar toda clase de errores:
 - Identificador mal escrito.
 - Números mal escritos.
 - Utilización de símbolos no permitidos.
 - Etc.

- **Prueba**
 - Se deberá comprobar el funcionamiento del analizador léxico usando tres ficheros:
 - Fichero denominado Newton.txt
 - ejemplo_1.txt: fichero original **sin** errores.
 - ejemplo_2.txt: fichero original **con** errores.