



# 4º Ingeniería Informática

## II26 Procesadores de lenguaje

Control de la práctica de microcalc (21 de noviembre de 2009)

### INSTRUCCIONES:

- La duración del examen es de dos horas.
- Antes de empezar, asegúrate de que el usuario con el que estás trabajando coincide con el del recuadro del final de este enunciado.
- Rellena el recuadro con tus datos.
- Ejecuta el programa `./prepara.py` pasándole como parámetro tu DNI. Este creará un fichero con tu DNI y nombre y un directorio llamado `microcalc`, donde debes guardar tu intérprete con sus modificaciones.
- El fichero principal debe llamarse `microcalc` y ser ejecutable.
- Al introducir el USB, debería montarse automáticamente.
- Utiliza el *script* `verifica.sh` de tu directorio *home* para comprobar que la estructura es la que pedimos. **Se penalizarán los exámenes para los que este script dé errores.**
- Cuando termines el examen, entrégnanos esta hoja.

### EJERCICIO

(0,5 PUNTOS)

Modifica tu implementación de `microcalc` para que acepte el nuevo operador sufijo (`-->`). Este operador es binario, infijo, asociativo por la izquierda, más prioritario que la suma y la resta y menos que la multiplicación y la división. El operando izquierdo debe ser de tipo cadena y el derecho de tipo entero. El resultado es de tipo cadena y contiene el sufijo de la cadena de la longitud indicada por el entero. Si el entero es mayor que la longitud de la cadena, el resultado es toda la cadena. Si el entero es negativo, se produce un error de ejecución. Algunos ejemplos de su uso son:

Expresión	Salida
"Hola Mundo" --> 5	Mundo
"Hola Mundo" --> -5	Linea 2: Error de ejecucion.
"Hola Mundo" --> --5	Mundo
"Hola \"Mundo\"" --> 3	do"
"Hola " + "Mundo" --> 7	Hola Mundo
("Hola " + "Mundo") --> 7	a Mundo
"Hola Mundo" --> 2*2	undo
("Hola Mundo" --> 2)*2	dodo
"Hola Mundo" --> 2*2 --> 6	undo

En el nivel sintáctico, tendrás que modificar la gramática. Te sugerimos que uses la siguiente:

$\langle \text{Línea} \rangle \rightarrow \langle \text{Expresión} \rangle \text{nl}$   
 $\langle \text{Expresión} \rangle \rightarrow \langle \text{Sufijo} \rangle (\text{opad } \langle \text{Sufijo} \rangle)^*$   
 $\langle \text{Sufijo} \rangle \rightarrow \langle \text{Término} \rangle (\text{opsuf } \langle \text{Término} \rangle)^*$   
 $\langle \text{Término} \rangle \rightarrow \langle \text{Factor} \rangle (\text{opmul } \langle \text{Factor} \rangle)^*$   
 $\langle \text{Factor} \rangle \rightarrow \text{entero} \mid \text{cadena} \mid \text{apar } \langle \text{Expresión} \rangle \text{ cpar} \mid \text{opad } \langle \text{Factor} \rangle \mid \text{barra } \langle \text{Expresión} \rangle \text{ barra}$

donde **opsuf** es la categoría léxica que representa el operador sufijo.

Recuerda que puedes obtener los  $n$  últimos caracteres de una cadena utilizando la notación de *slices* de Python. Por ejemplo "Hola Mundo" `[-5:]` devuelve `Mundo`.