

## Grupo Jueves 12:00-14:00 semanas A

### - Práctica 3 -

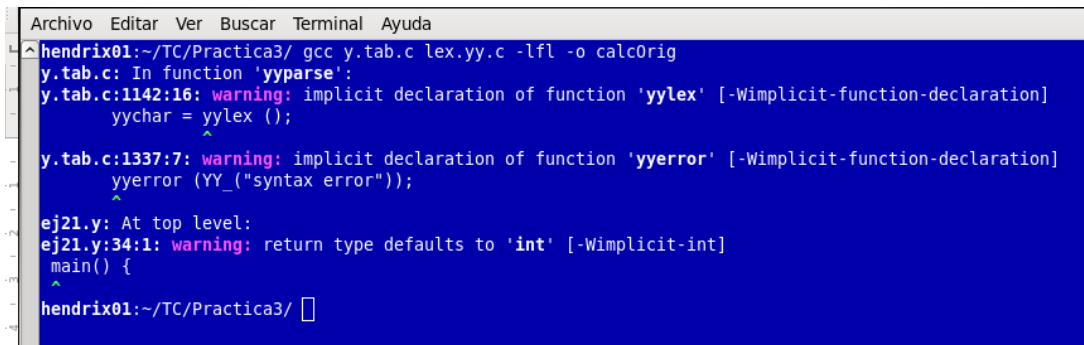
**Autor:** Jesús Aguas Acín

## EJERCICIO 1:

### 1. Indica a qué gramática corresponde calcOrig.y

calclist → calclistexpEOL | ε  
exp → factor | exp+factor | exp-factor  
factor → term | exp\*factor | exp/factor  
term → [0-9] | (exp)

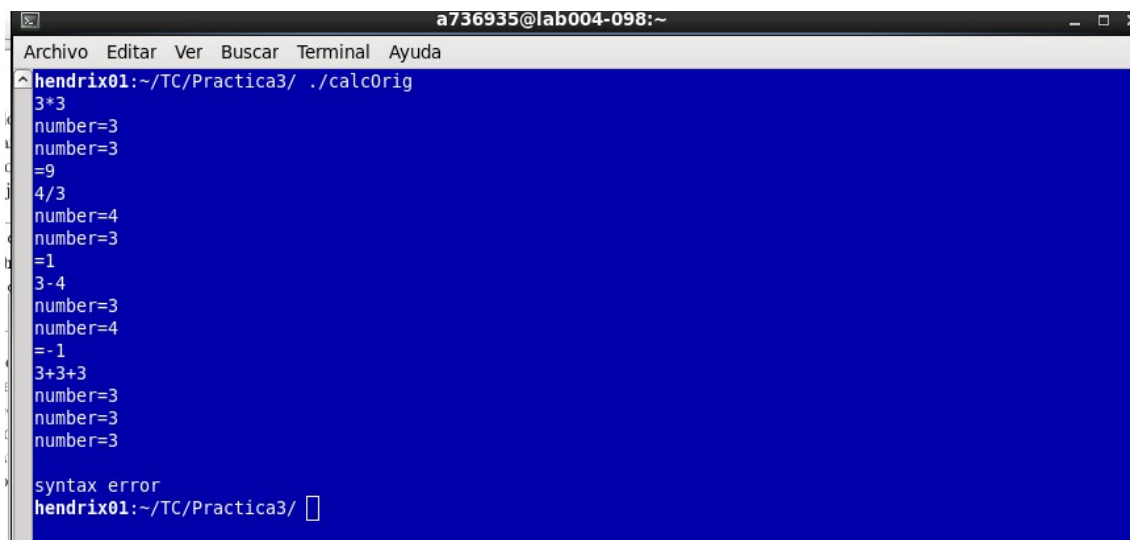
### 2. Mensajes de error



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
^hendrix01:~/TC/Practica3/ gcc y.tab.c lex.yy.c -lfl -o calcOrig
y.tab.c: In function 'yyparse':
y.tab.c:1142:16: warning: implicit declaration of function 'yylex' [-Wimplicit-function-declaration]
    yychar = yylex ();
                  ^
y.tab.c:1337:7: warning: implicit declaration of function 'yyerror' [-Wimplicit-function-declaration]
    yyerror (YY_("syntax error"));
    ^
ej21.y: At top level:
ej21.y:34:1: warning: return type defaults to 'int' [-Wimplicit-int]
    main() {
    ^
hendrix01:~/TC/Practica3/
```

Salen 3 warnings, salen siempre al compilar archivos bison.

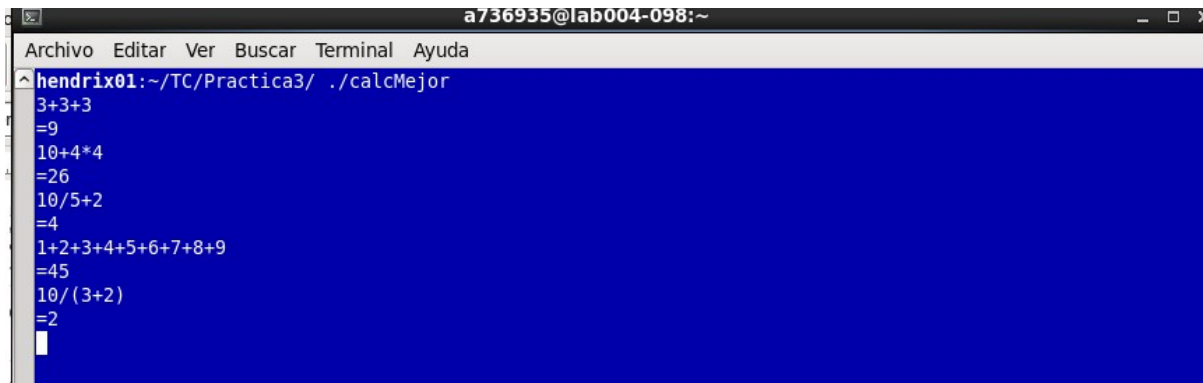
### 3. Probando calcOrig



```
a736935@lab004-098:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
^hendrix01:~/TC/Practica3/ ./calcOrig
3*3
number=3
number=3
=9
4/3
number=4
number=3
=1
3-4
number=3
number=4
=-1
3+3+3
number=3
number=3
number=3
syntax error
hendrix01:~/TC/Practica3/
```

La calculadora no funciona en el caso de que haya más de dos operandos.

#### 4. calcOrig vs. calcMejor



```
a736935@lab004-098:~
Archivo Editar Ver Buscar Terminal Ayuda
^hendrix01:~/TC/Practica3/ ./calcMejor
3+3
=9
10+4*4
=26
10/5+2
=4
1+2+3+4+5+6+7+8+9
=45
10/(3+2)
=2
```

La calculadora calcMejor soluciona los problemas encontrados en calcOrig, ahora se puede operar con todos los operandos que se necesiten y respeta las propiedades de la jerarquía de operaciones. Esto se debe a que se incluyen mejoras en el fichero bison calcMejor.y, donde incluimos variables no terminales que permiten tener el resultado de una operación en una misma variable de modo que podemos utilizarla para realizar más operaciones. Como resultado podemos operar con múltiples operandos.

#### EJERCICIO 2:

##### 1. Resumen

Este ejercicio se divide en 3 subapartados:

**2.1:** En esta modificación de calcMejor basta únicamente con modificar el archivo flex, de modo que reconozca como patrón los números binarios y pasar dicho número a decimal mediante una función en código C. Ya con el número en decimal se opera de forma normal.

**2.2:** Para que la nueva versión de calcMejor cumpla con la especificación debemos modificar primero el archivo flex, de modo que reconozca el “;” al final de la línea en el caso de que no sea binario y “;b” en el caso de que queramos que el resultado sea en binario. En el fichero bison deberemos implementar una función que pase un número en decimal (el resultado de la operación) a binario. Y utilizar dicha función para transformar el resultado de la operación, en el caso de que se haya especificado al final de la operación (;b).

**2.3:** En este apartado el objetivo es crear una variable acum , capaz de ser inicializada con un valor o una expresión y que se pueda utilizar la misma para operar. Para ello lo primero diferenciamos entre el patrón de inicialización de la variable(acum:=) del uso de la misma para operar (acum) en el archivo flex, según sea uno u otro lo utilizaremos de una manera distinta. Para poder utilizar el acum como un operando basta con incluirlo como una posible expresión o factor simple. Para el caso de la inicialización basta con crear una variable entera acum e inicializarla con la expresión que se indique, seguido del fin de línea.

### EJERCICIO 3:

Para implementar la gramática del enunciado debemos crear 3 tokens “x”, “y”, “z” y 3 variables no terminales “S”, “B”, “C” que cumplirán las reglas descritas en el enunciado. El fichero flex tiene que reconocer las “x,y,z” y devolver al fichero bison el token correspondiente.

La expresión regular que reconoce la gramática es:  $(xxz(\epsilon+y)xx)^*(zx)^*$

La he obtenido creando varias cadenas reconocidas por la gramática y tratando de reconocer el patrón que se cumplía en todas ellas.