



Implementar en Java mediante la técnica LL(1) un programa que realice el análisis sintáctico de una lista de sentencias de un lenguaje imperativo similar a C, que se define mediante la siguiente gramática:¹

R ₀	axioma	-> listaSent EOF
R ₁	listaSent	-> sent listaSent
R ₂	listaSent	->
R ₃	sent	-> WHILE AP cond AP sent
R ₄	sent	-> DO sent WHILE AP cond CP PYC
R ₅	sent	-> IDENT IGUAL var PYC
R ₆	sent	-> ALL listaSent CLL
R ₇	cond	-> var MENOR var
R ₈	var	-> IDENT
R ₉	var	-> NUMERO

Para ello, se proporciona un analizador léxico implementado mediante *JFlex*, en el fichero `Lex.lex`, la clase `Yytoken.java` que contiene la definición de los tokens y las funciones auxiliares para la salida;² y un esquema de la clase `Parser.java`, que contiene el método `main`. Ésta última es la que debe entregarse una vez completada.

Este analizador tendrá como entrada un programa que con una lista de sentencias y como salida la secuencia de reglas. Se proporcionan diversos ejemplos de entrada y salida, y un programa compilado que realiza la misma función que debe realizar el programa que se pide implementar.

Para compilar y ejecutar el programa se usarán la secuencia

```
jflex Lex.lex
javac Parser.java
java Parser prog.in
```

En donde `Lex.lex` es el fichero que contiene el analizador léxico, `Parser.java` el fichero con el analizador sintáctico, (a completar por el alumno), `prog.in` es el fichero de entrada y `prog.out` el fichero de salida.

Para probar la entrada y la salida, puede usarse el programa pre-compilado, al cual hay previamente que dar permiso de ejecución

```
chmod +x parser
./parser prog.in
```

¹ Para definir esta gramática se han empleado identificadores en minúscula para los símbolos no-terminales, e identificadores en mayúscula para los símbolos terminales. Las reglas cuyo consecuente es ϵ muestran el consecuente vacío.

² Para garantizar que la salida tiene un formato correcto y facilitar la corrección automática de este ejercicio, deben usarse las funciones `Yytoken.regla(int)` y `YYtoken.error(int)` para escribir las reglas y los posibles errores respectivamente..