

Construir un analizador de código de C++ o C, en el cual podamos identificar por pantalla rápidamente las librerías, variables, funciones y operadores usados en el código. Además también debe ser capaz de reconocer si las expresiones usadas son correctas, esto quiere decir que el uso de los paréntesis y corchetes en el código sea correcto.

Para este problema que hemos resuelto en lex, hemos hecho una serie de reglas para identificar si lo que estamos leyendo son librerías (las cuales metemos directamente al vector), variables o funciones al igual que los tipos de estas, las cuales luego las hemos pasado a una función donde procesamos el texto y lo metemos en un set, en el cual ya separamos el tipo de la variable o función y el nombre.

Para reconocer si las expresiones son correctas hacemos un contador de paréntesis y corchetes, tanto abiertos como cerrados, luego restamos el número de corchetes abiertos con cerrados y el número de (paréntesis abiertos + número de funciones) con el número de paréntesis cerrados y si ambos son iguales a 0 la expresión será correcta. Sumar las funciones con el número de paréntesis es necesario para el correcto funcionamiento, ya que para identificar funciones debemos leer "TIPO NOMBRE(", ese paréntesis se lo traga y no lo suma, por lo que a la hora de comprobar si la expresión es correcta deberemos sumarlo de esta manera.

Para comprobar el correcto funcionamiento hemos hecho un pequeño código de prueba en C++, el cual introducimos por terminal a la hora de ejecutar nuestro analizador de código.

Dejamos unas imágenes de la compilación y funcionamiento de nuestro analizador de código en Lex:

```
manuuarizaa@manuuarizaa-Aspire-E5-771G:~/Dropbox/INFORMATICA CLASE/Tercero/1erCu  
atrimestre/MC/MCMIERCOLES/VersionFinal$ lex trabajolex.l  
manuuarizaa@manuuarizaa-Aspire-E5-771G:~/Dropbox/INFORMATICA CLASE/Tercero/1erCu  
atrimestre/MC/MCMIERCOLES/VersionFinal$ g++ lex.yy.c -o comprobador -ll  
manuuarizaa@manuuarizaa-Aspire-E5-771G:~/Dropbox/INFORMATICA CLASE/Tercero/1erCu  
atrimestre/MC/MCMIERCOLES/VersionFinal$ ./comprobador prueba.cpp
```

```
*****ANALISIS DEL CODIGO*****
La expresion es correcta

Los operadores son:
+ ha aparecido 3 veces
- ha aparecido 1 veces
* ha aparecido 1 veces
/ ha aparecido 1 veces
++ ha aparecido 3 veces
-- ha aparecido 1 veces

Se han usado las bibliotecas:
#include<iostream>
#include<cmath>

Existen 16 variables.
Tipo: bool      Nombre: lohace
Tipo: char      Nombre: esodice
Tipo: double    Nombre: d1
Tipo: double    Nombre: d2
Tipo: float     Nombre: nlargo
Tipo: int       Nombre: i
Tipo: int       Nombre: n1
Tipo: int       Nombre: n2
Tipo: int       Nombre: n3
Tipo: int       Nombre: numero1
Tipo: int       Nombre: numero2
Tipo: int       Nombre: numero3
Tipo: int       Nombre: numero4
Tipo: string    Nombre: esodicelargo
Tipo: unsigned  Nombre: amitampoco
Tipo: unsigned  Nombre: nuncameusan

Existen 3 funciones.
Tipo: bool      Nombre: sehasumado
Tipo: int       Nombre: main
Tipo: int       Nombre: sumar
```

Adjuntamos el código de prueba usado al igual que el código de nuestro programa Lex junto con este documento.