



ESCUELA POLITÉCNICA NACIONAL

Compiladores y Lenguajes

PARALELO: GR1CC

TAREA

Ejercicios Lex

INTEGRANTES:

Mark Hernández

Ozzy Loachamín

Luis Pérez

Sebastián Zambonino

2024-A

PROGRAMA QUINCE

(quince.l)

(enunciado del ejercicio si es que hay)

DESCRIPCIÓN

Escáner para un lenguaje al estilo de Pascal.

CÓDIGO

```
%{
/* Se necesita esto para la llamada a atof() más abajo */
#include <math.h>
}%

DIGITO    [0-9]
ID         [a-z][a-z0-9]*

%%

{DIGITO}+  {
    printf( "Un entero: %s (%d)\n", yytext,
    atoi( yytext ) );
}

{DIGITO}+"."{DIGITO}*  {
    printf( "Un real: %s (%g)\n", yytext,
    atof( yytext ) );
}

if|then|begin|end|procedure|function  {
    printf( "Una palabra clave: %s\n", yytext );
}

{ID}      printf( "Un identificador: %s\n", yytext );
"+"|"-"|"*"|"/"  printf( "Un operador: %s\n", yytext );
"{"["^}\n]*"      /* se come una línea de comentarios */
[ \t\n]+         /* se come los espacios en blanco */
.    printf( "Carácter no reconocido: %s\n", yytext );
```

```

%%
int main(int argc, char **argv )
{
    ++argv, --argc; /* se salta el nombre del programa */
    if ( argc > 0 )
        yyin = fopen( argv[0], "r" );
    else
        yyin = stdin;

    yylex();
}

```

La función main tiene que ser especificada el tipo de datos, junto con argc, y argv, al menos para la versión de c que se utiliza.

EJECUCIÓN

```

● kyk@Muguiwara:~/Universidad/compiladores/bimestre2/lex$ cd quince/
● kyk@Muguiwara:~/Universidad/compiladores/bimestre2/lex/quince$ lex quince.l
● kyk@Muguiwara:~/Universidad/compiladores/bimestre2/lex/quince$ gcc lex.yy.c -lfl
quince.l:28:1: warning: return type defaults to 'int' [-Wimplicit-int]
 28 | main( argc, argv )
    | ~~~~
● kyk@Muguiwara:~/Universidad/compiladores/bimestre2/lex/quince$ lex quince.l
● kyk@Muguiwara:~/Universidad/compiladores/bimestre2/lex/quince$ gcc lex.yy.c -lfl
○ kyk@Muguiwara:~/Universidad/compiladores/bimestre2/lex/quince$ ./a.out
25
Un entero: 25 (25)
33.55
Un real: 33.55 (33.55)
if
Una palabra clave: if
then
Una palabra clave: then
epn2020
Un identificador: epn2020
23*10
Un entero: 23 (23)
Un operador: *
Un entero: 10 (10)
/*
Un operador: /
Un operador: *
@
Carácter no reconocido: @

```