

## ☐ Kevin Badillo Olmos 6-3

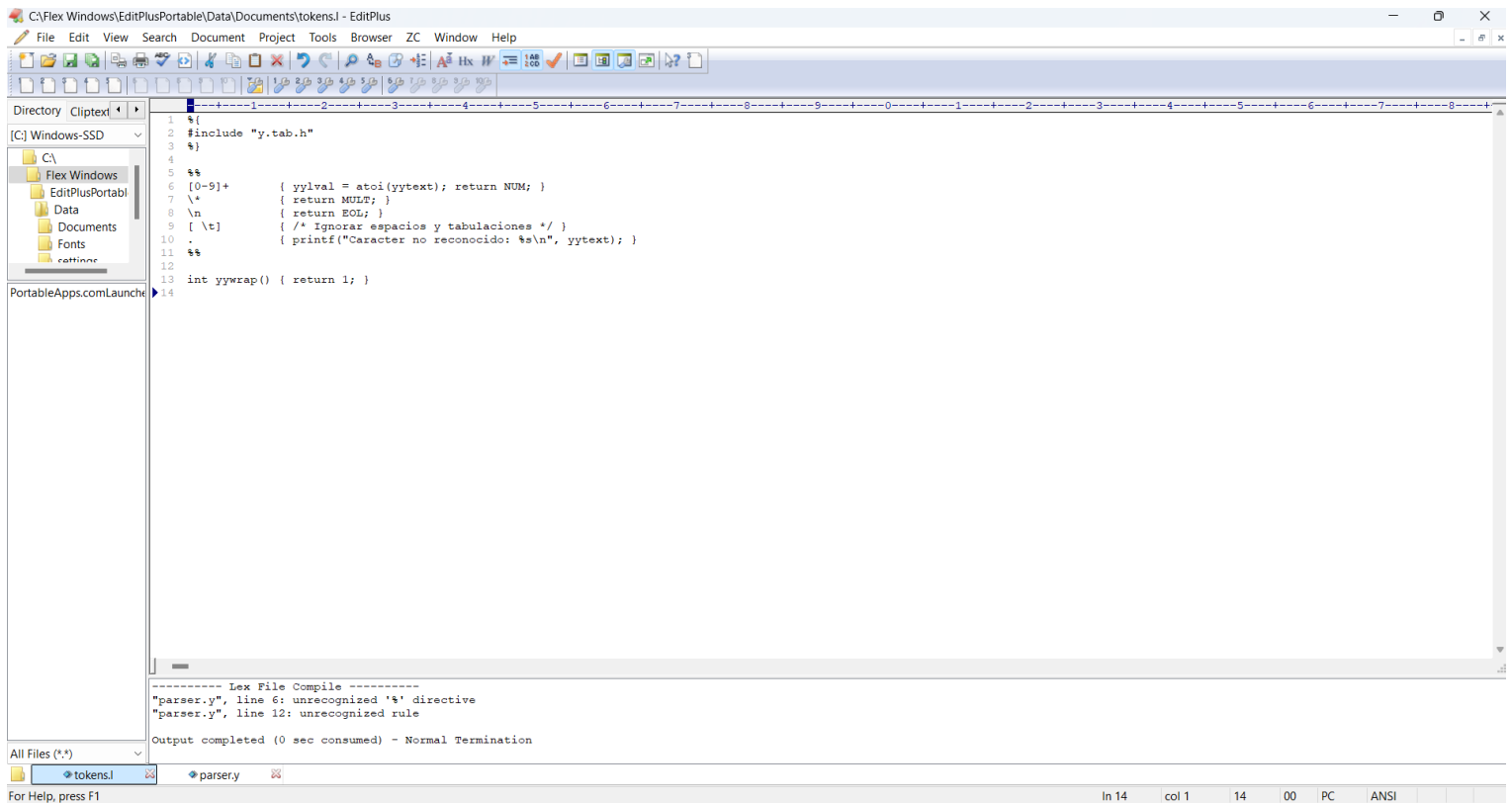
Genera un programa de lex para reconocer los tokens. a) números  
b) operador aritmético de multiplicación: \*

☐ Genera el programa en yacc para verificar que una línea de código donde se exprese la multiplicación de dos números está correctamente construida.

☐ Ejecuta los dos programas para verificar la siguiente entrada: 133\*25

☐ Entrega el enlace al documento donde se encuentren los códigos y las pantallas de ejecución con los resultados.

## LEX



```
1 %{
2 #include "y.tab.h"
3 %}
4
5 **
6 [0-9]+ { yylval = atoi(yytext); return NUM; }
7 \* { return MULT; }
8 \n { return EOL; }
9 [ \t] { /* Ignorar espacios y tabulaciones */ }
10 . { printf("Caracter no reconocido: %s\n", yytext); }
11 **
12
13 int yywrap() { return 1; }
14
```

----- Lex File Compile -----  
"parser.y", line 6: unrecognized '\*' directive  
"parser.y", line 12: unrecognized rule  
Output completed (0 sec consumed) - Normal Termination

# Yacc

```
1  %{
2  #include <stdio.h>
3  #include <stdlib.h>
4  %}
5
6  %token NUM MULT EOL
7
8  %%
9  linea:
10     expr EOL { printf("La línea de código es válida.\n"); }
11     | EOL { /* Línea vacía */ }
12     ;
13
14  expr:
15     NUM MULT NUM { printf("Resultado: %d\n", $1 * $3); }
16     ;
17  %%
18
19  int main() {
20     printf("Ingrese una línea de código: ");
21     yyparse();
22     return 0;
23 }
24
25 int yyerror(char *s) {
26     fprintf(stderr, "Error: %s\n", s);
27     return 0;
28 }
29
```

----- Lex File Compile -----  
"parser.y", line 6: unrecognized '%' directive  
"parser.y", line 12: unrecognized rule  
Output completed (0 sec consumed) - Normal Termination

For Help, press F1

In 29 col 1 29 00 PC ANSI