Expt. No. 2 Page No. 4

Expt. Name. Use LEX and YACC to implement parser on ambigous grammar. Date :

Aim:

To write the program using LEX and YACC to implement parser on ambigous grammar.

Algorithm: → File.l

step 1: start

step 2: Include the necessary header files and declare the necessary variables.

step 3: initialize the digits, operators, parenthesis and return the value else print Syntax Error

step 4: Call the function & return I

step 5: Stop

File.y

step 1: start

step 2: Include the necessary header files and declare the necessary variables

step 3: Substitute the values and Calculate respectively for Addition, Subtraction, Multiplication and division and return the result.

step 4: Call the main function and print the result

step 5: stop.

Expt. No. _____ Page No. 5

Expt. Name. _____ Date : _____

Program:

File.l

```
%option noyywrap
%{
    #include<stdio.h>
    #include "y.tab.h"
    void yyerror(char *s)
    extern int yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext);
return NUM;
[-+*/\n]  {return *yytext;}
"("       {return *yytext;}
")"       {return *yytext;}
[\t];
.         {yyerror("Syntax Error");}
%%
int yywrap()
 {
 return 1;
 }
```

File.y

```
%{
    #include<stdio.h>
    extern int yylex(void);
    void yyerror(char *);
%}
```

Expt. No. _____ Page No. 6

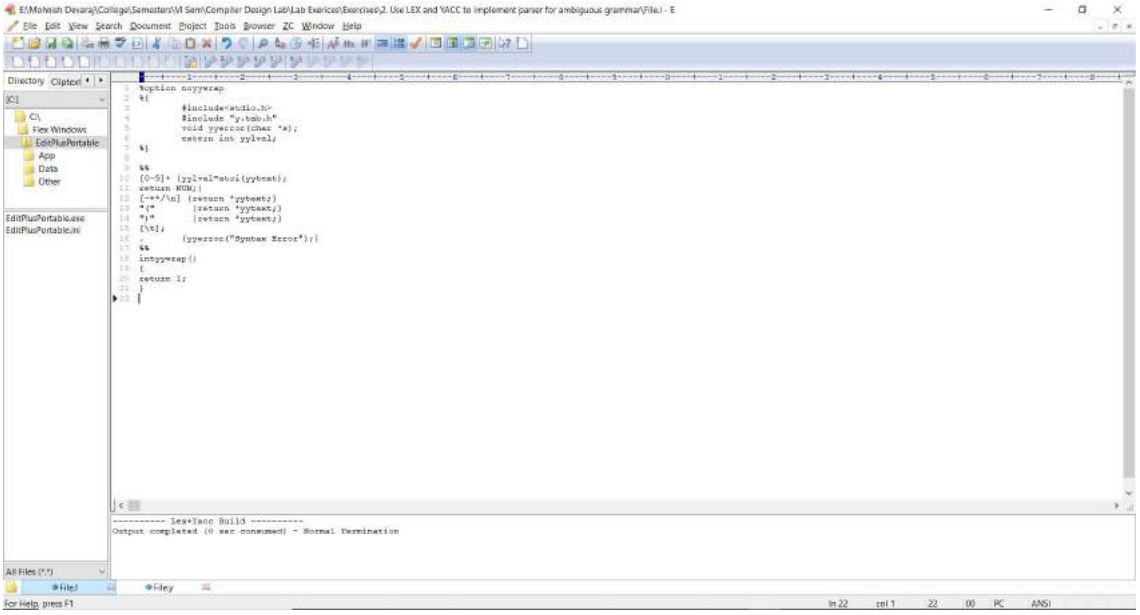
Expt. Name. _____ Date : _____

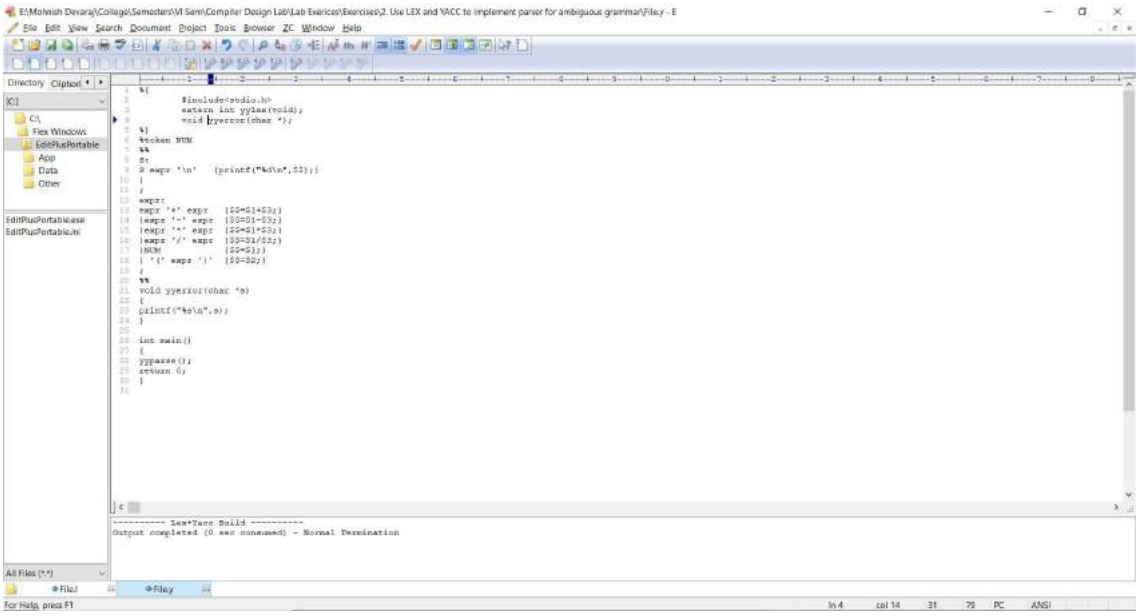
```
%token NUM
%%
S:
S expr '\n' {printf("%d\n",$2);}
|
;
expr:
expr '+' expr    {$$ = $1 + $3;}
|expr '-' expr    {$$ = $1 - $3;}
|expr '*' expr    {$$ = $1 * $3;}
|expr '/' expr    {$$ = $1 / $3;}
| NUM            {$$ = $1;}
| '('expr')'     {$$ = $2;}
;
%%
void yyerror(char *s)
{
printf("%s\n",s);
}

int main()
{
yyparse();
return 0;}
```

Result:

Use LEX and YACC to implement parser for ambiguous is executed successfully.


E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exercises\Exercises\2. Use LEX and YACC to implement parser for ambiguous grammar\File.l - E
File Edit View Search Document Project Tools Browser ZC Window Help
Directory Cliptext
C:\
Flex Windows
EditPlusPortable
App
Data
Other
EditPlusPortable.exe
EditPlusPortable.ini
All Files (*.*)
File.l
File.y
%option noyywrap
%{
#include<stdio.h>
#include "y.tab.h"
void yyerror(char *s);
extern int yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext);
return NUM;}
[-+*/\n] {return *yytext;}
"("
{return *yytext;}
")"
{return *yytext;}
[\t];
.
{yyerror("Syntax Error");}
%%
intyywrap()
{
return 1;
}
---------- Lex+Yacc Build ----------
Output completed (0 sec consumed) - Normal Termination
For Help, press F1
ln 22
col 1
22
00
PC
ANSI


E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exercises\Exercises\2. Use LEX and YACC to implement parser for ambiguous grammar\File.y - E
File Edit View Search Document Project Tools Browser ZC Window Help
Directory Cliptext
C:\
Flex Windows
EditPlusPortable
App
Data
Other
EditPlusPortable.exe
EditPlusPortable.ini
All Files (*.*)
File.l
File.y
%{
#include<stdio.h>
extern int yylex(void);
void yyerror(char *);
%}
%token NUM
%%
S:
S expr '\n'
{printf("%d\n",$1);}
|
;
expr:
expr '+' expr
{$$=$1+$3;}
|expr '-' expr
{$$=$1-$3;}
|expr '*' expr
{$$=$1*$3;}
|expr '/' expr
{$$=$1/$3;}
|NUM
{$$=$1;}
| '(' expr ')'
{$$=$2;}
;
%%
void yyerror(char *s)
{
printf("%s\n",s);
}
int main()
{
yyparse();
return 0;
}
---------- Lex+Yacc Build ----------
Output completed (0 sec consumed) - Normal Termination
For Help, press F1
ln 4
col 14
31
79
PC
ANSI


E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exerices\Exercises\2. Use LEX and YACC to implement pars...
2+3
5
4+8
12
12/3
4