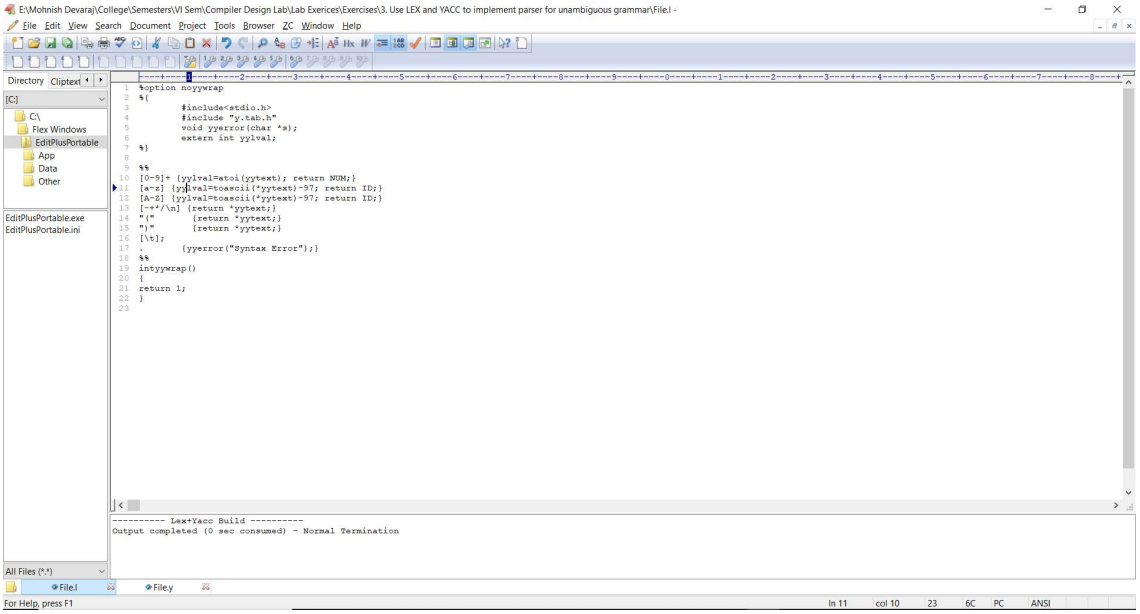
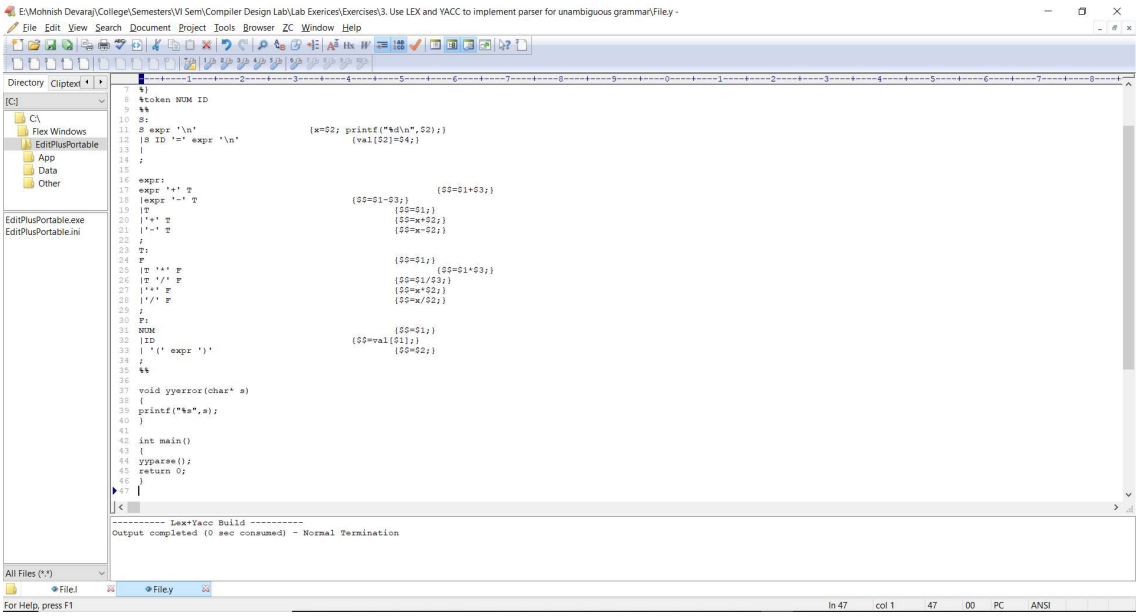
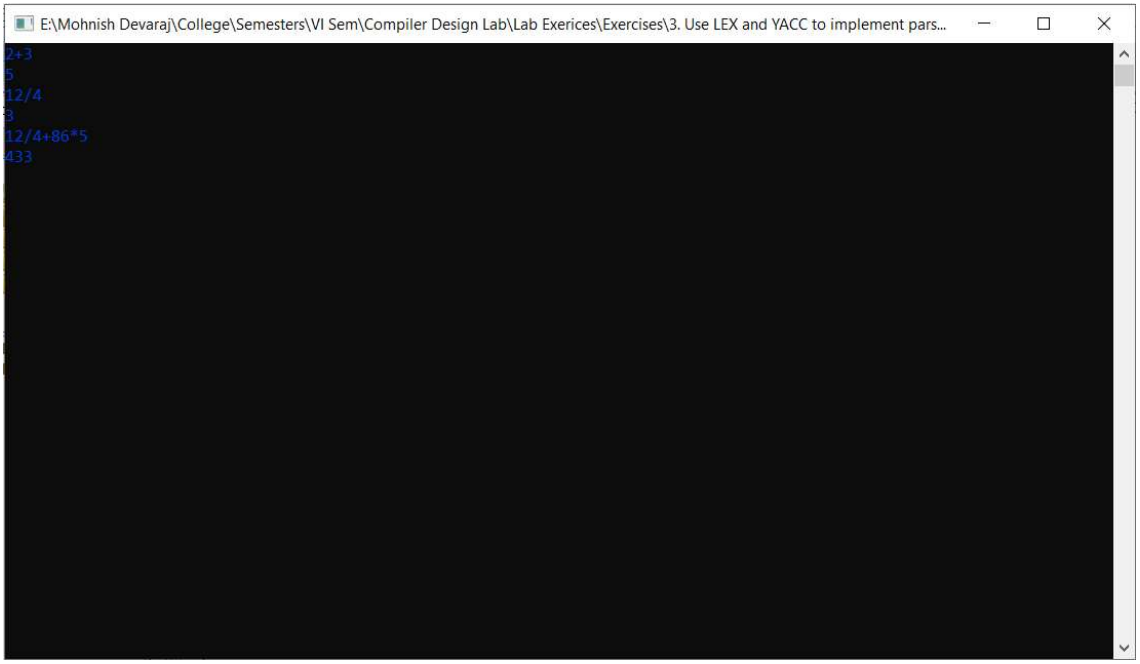

E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exercices\Exercises\3. Use LEX and YACC to implement parser for unambiguous grammar\File.l -
File Edit View Search Document Project Tools Browser ZC Window Help
Directory Cliptext
[C:]
C:\
Flex Windows
EditPlusPortable
App
Data
Other
EditPlusPortable.exe
EditPlusPortable.ini
All Files (*.*)
%option noyywrap
%{
#include<stdio.h>
#include "y.tab.h"
void yyerror(char *s);
extern int yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext); return NUM;}
[a-z] {yylval=toascii(*yytext)-97; return ID;}
[A-Z] {yylval=toascii(*yytext)-97; return ID;}
[-+*/\n] {return *yytext;}
"(" {return *yytext;}
")" {return *yytext;}
[\t];
. {yyerror("Syntax Error");}
%%
intyywrap()
{
return 1;
}
---------- Lex+Yacc Build ----------
Output completed (0 sec consumed) - Normal Termination
File.l
File.y
For Help, press F1
ln 11
col 10
23
6C
PC
ANSI


E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exercices\Exercises\3. Use LEX and YACC to implement parser for unambiguous grammar\File.y -
File Edit View Search Document Project Tools Browser ZC Window Help
Directory Cliptext
[C:]
C:\
Flex Windows
EditPlusPortable
App
Data
Other
EditPlusPortable.exe
EditPlusPortable.ini
All Files (*.*)
%}
%token NUM ID
%%
S:
S expr '\n' {x=$2; printf("%d\n",$2);}
|S ID '=' expr '\n' {val[$2]=$4;}
|
;
expr:
expr '+' T {$$=$1+$3;}
|expr '-' T {$$=$1-$3;}
|T {$$=$1;}
|'+' T {$$=x+$2;}
|'-' T {$$=x-$2;}
;
T:
F {$$=$1;}
|T '*' F {$$=$1*$3;}
|T '/' F {$$=$1/$3;}
|'*' F {$$=x*$2;}
|'/' F {$$=x/$2;}
;
F:
NUM {$$=$1;}
|ID {$$=val[$1];}
| '(' expr ')' {$$=$2;}
;
%%
void yyerror(char* s)
{
printf("%s",s);
}
int main()
{
yyparse();
return 0;
}
---------- Lex+Yacc Build ----------
Output completed (0 sec consumed) - Normal Termination
File.l
File.y
For Help, press F1
ln 47
col 1
47
00
PC
ANSI


E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exerices\Exercises\3. Use LEX and YACC to implement pars...
2+3
5
12/4
3
12/4+86*5
433