2CS701 Compiler Construction

Lab-5 Task

Submitted by: Labdhi Sheth 18BCE101

Aim: To implement a calculator in YACC

Code:

calculator.l

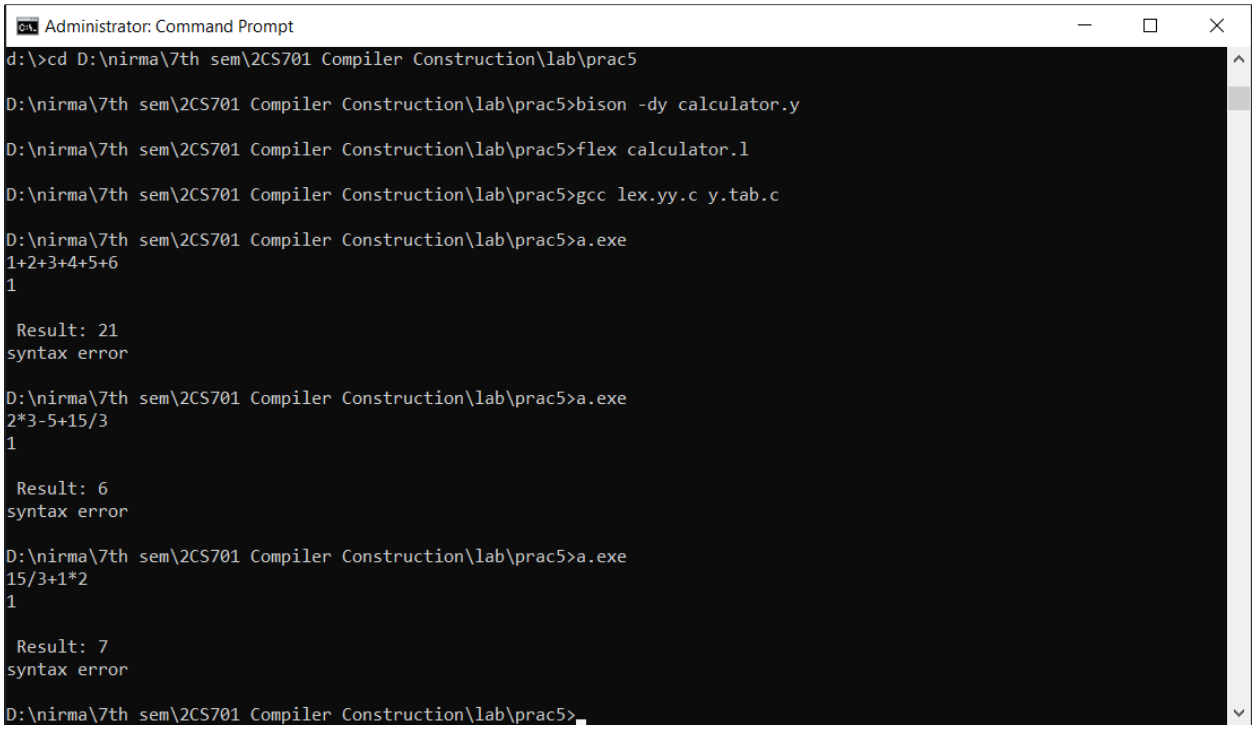
```
%{
    #include<stdio.h>
    #include<stdlib.h>
    #include "y.tab.h"
%}

%%
[0-9]+ {yylval.a_number = atoi(yytext); return number;}
[-+*/();] {return yytext[0];}
[ \t\n] {;}
. {ECHO; yyerror("Not Valid!");}

%%

int yywrap(void)
{
    return 0;
}
```

calculator.y

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

%union {int a_number;}
%start line
%token <a_number> number
%type <a_number> exp term factor

%%
line  : exp {printf("\n Result: %d \n", $1);};
exp   : term {$$ = $1;} | exp'+'term {$$=$1+$3;} | exp'-'term
{$$=$1-$3;};
term  : factor {$$ = $1;} | term'*'factor {$$=$1*$3;} |
term'/'factor {$$=$1/$3;};
factor  : number {$$=$1;} | '('exp')' {$$=$2;} | '-'factor
{$$=-$2;};
%%

int main(void) {return yyparse();}
void yyerror(char *s){fprintf(stderr, "%s\n", s);}
```

Output:

Name	Date modified	Type	Size
a	17-10-2021 07:28 PM	Application	27 KB
calculator.l	17-10-2021 07:26 PM	L File	1 KB
calculator.y	17-10-2021 07:26 PM	Y File	1 KB
lex.yy	17-10-2021 07:27 PM	C Source File	37 KB
y.tab	17-10-2021 07:27 PM	C Source File	44 KB
y.tab	17-10-2021 07:27 PM	C Header File	3 KB



Conclusion:

From this practical, we understood the concept of YACC. We understood how the tokens are generated from an arithmetic expression using the lexical analyzer and given to YACC to generate a parse tree to evaluate the arithmetic expression.