

Christian Mesina
Hugo Romero
Luis Escobar Urrutia

Lex/Yacc Simple Calculator Project

Introduction:

- For this project, we created a simple calculator program using Lex/Yacc. We got all the information and references from the 570 website and we used Andrew Brinker's code which was given to us as a starting point for our program. Our job was to add an exit/quit feature, so that we can enter one expression and then use the result as the start of another expression - for example:
2+5 => 7
/2 => 3.5

Source Code:

- **calc.yacc**
/*
* Name: Christian Mesina, Hugo Romero, Luis Escobar Urrutia
* Class: CSE 570 Compilers
* Instructor: Dr. Ernesto Gomez
* Term: Spring 2020
**/

%{
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define PI 3.14159265358979323846
int regs[26];
double answer;
int yylex();
int yyerror(char *s);
int yywrap();
%}

%start list
%union {
double a;
char c;
}
%type <a> expr number DIGIT
%type <c> LETTER
%token DIGIT LETTER
%token QUIT
%left '|' '
%left '&'

```

%left '+' '-'
%left '*' '/' '%'
%left UMINUS

%%

list: /* empty */
    | list stat '\n'
    | list error '\n' {
        yyerrok;
    };

stat: expr {
    printf("%f\n", $1);
}
| LETTER '=' expr {
    regs[$1] = $3;
};

expr: '(' expr ')' {
    $$ = $2;
    answer = $2;
}
| expr '*' expr {
    $$ = $1 * $3;
    answer = $$;
}
| '*' expr {
    $$ = answer * $2;
    answer = $$;
}
| expr '/' expr {
    $$ = $1 / $3;
    answer = $$;
}
| '/' expr {
    $$ = answer / $2;
    answer = $$;
}
| expr '+' expr {
    $$ = $1 + $3;
    answer = $$;
}
| '+' expr {
    $$ = answer + $2;
    answer = $$;
}
| expr '-' expr {
    $$ = $1 - $3;
    answer = $$;
}

```

```

    }
    | '-' expr {
        $$ = answer - $2;
        answer = $$;
    }
    | '-' expr %prec UMINUS {
        $$ = -$2;
        answer = $$;
    }
    | LETTER {
        $$ = regs[$1];
    }
    | number;
    | QUIT {
        return EXIT_SUCCESS;
    }
}

number: DIGIT {
    $$ = $1;
};

%%

int main() {
    printf("Enter an expression:\n");
    return yyparse();
}

int yyerror(char *s) {
    fprintf(stderr, "%s\n", s);
    return 1;
}

int yywrap() {
    return 1;
}

```

- **calc.lex**

```
/*
 * Name: Christian Mesina, Hugo Romero, Luis Escobar Urrutia
 * Class: CSE 570 Compilers
 * Instructor: Dr. Ernesto Gomez
 * Term: Spring 2020
 **/

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

int c;
double d;
extern YYSTYPE yylval;
%}

%%

" ";

[a-z] {
    c = yytext[0];
    yylval.a = c - 'a';
    return(LETTER);
}

[0-9] {
    d = atof(yytext);
    yylval.a = d - 0;
    return(DIGIT);
}

[^a-z0-9\b] {
    c = yytext[0];
    return(c);
}

[0-9]+\.[0-9]+ {
    d = atof(yytext);
    yylval.a = d - 0;
    return(DIGIT);
}

QUIT {
    return QUIT;
}
```

Outputs:

```
[005319687@csusb.edu@jlb359-2 Project]$ lex calc.lex
[005319687@csusb.edu@jlb359-2 Project]$ yacc -d calc.yacc
yacc: 4 reduce/reduce conflicts.
[005319687@csusb.edu@jlb359-2 Project]$ cc lex.yy.c y.tab.c -o project
[005319687@csusb.edu@jlb359-2 Project]$ ./project
Enter an expression:
1+1
2.000000
*3
6.000000
/6
1.000000
-1
0.000000
QUIT
[005319687@csusb.edu@jlb359-2 Project]$ ./project
Enter an expression:
2+5
7.000000
/2
3.500000
QUIT
[005319687@csusb.edu@jlb359-2 Project]$
```