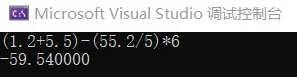
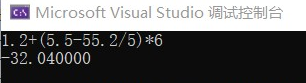
程序设计2实验报告

张志博 2017211416

1. 实验题目
   1. 语法分析程序的设计与实现。
2. 实验内容
   1. 编写语法分析程序，实现对算术表达式的语法分析。要求所分析算数表达式由如下的文法产生。
   2. 文法
      1. E → E+T | E-T | T
      2. T → T\*F | T/F | F
      3. F → (E) | num
3. 实验要求
   1. 在对输入的算数表达式进行分析的过程中，依次输出所采用的产生式。用以下四种方法之一实现。
   2. 方法
      1. 编写递归调用程序实现自顶向下的分析。
      2. 编写LL(1)语法分析程序，要求如下。
         1. 编程实现算法4.2，为给定文法自动构造预测分析表。
         2. 编程实现算法4.1，构造LL(1)预测分析表。
      3. 编写语法分析程序实现自底向上的分析，要求如下。
         1. 构造识别该文法所有活前缀的DFA。
         2. 构造该文法的LR分析表。
         3. 编程实现算法4.3，构造LR分析程序。
      4. 利用YACC自动生成语法分析程序，调用LEX自动生成的词法分析程序。
4. 实验平台与环境
   1. Windows 10
   2. Visual Studio 2019
   3. Flex + Bison
5. 实验分析
   1. 环境配置
      1. 下载win\_flex\_bison-latest。
      2. 向VS项目中添加win\_flex\_bison软件中的targets文件所在的路径。
         1. 工程->生成依赖项->生成自定义->查找现有的->\win\_flex\_bison-latest\custom\_build\_rules\win\_flex\_bison\_custom\_build.targets。
      3. 向VS项目中添加win\_bison.exe和win\_flex.exe文件所在的路径。
         1. 工程->属性->VC++目录->可执行文件目录->\win\_flex\_bison-latest\
      4. 编译和运行lex和yacc文件，再把生成的demo.flex.cpp、demo.tab.cpp、demo.tab.h添加到VS项目中，重新编译并运行即可。
   2. 项目结构
      1. LEX编译程序根据LEX源程序产生的词法分析程序yylex()可用于YACC，名字yylex()就是YACC所需要的词法分析程序的名字。函数yylex()可以使用YACC中定义的记号，因为LEX的输出是YACC输出文件的一部分，所以每个LEX动作都返回YACC知道的终结符号。
      2. 根据词法写出LEX源程序demo.l，生成demo.flex.cpp。
         1. **%{**
         2. **// Declaration section**
         3. **#include <stdlib.h>**
         4. **#define YYSTYPE double**
         5. **#include "demo.tab.h"**
         6. **%}**
         7. **ws [ \t]+**
         8. **digit [0-9]**
         9. **num {digit}+(\.{digit}+)?**
         10. **%%**
         11. **{ws} {}**
         12. **{num} {yylval = atof(yytext); return NUM;}**
         13. **"+" {return '+';}**
         14. **"-" {return '-';}**
         15. **"\*" {return '\*';}**
         16. **"/" {return '/';}**
         17. **"(" {return '(';}**
         18. **")" {return ')';}**
         19. **"\n" {return '\n';}**
         20. **%%**
         21. **int yywrap(void) {**
         22. **return 1;**
         23. **}**
      3. 根据文法写出YACC源程序demo.y，生成demo.tab.cpp和demo.tab.h。
         1. **%{**
         2. **#include <stdio.h>**
         3. **#include <ctype.h>**
         4. **#define YYSTYPE double**
         5. **extern int yylex();**
         6. **void yyerror(const char \*s);**
         7. **%}**
         8. **%token NUM**
         10. **%%**
         11. **line: E'\n' {printf("%lf\n", $1); return 0;}**
         12. **;**
         13. **E: E'+'T {$$ = $1 + $3;}**
         14. **| E'-'T {$$ = $1 - $3;}**
         15. **| T {$$ = $1;}**
         16. **;**
         17. **T: T'\*'F {$$ = $1 \* $3;}**
         18. **| T'/'F {$$ = $1 / $3;}**
         19. **| F {$$ = $1;}**
         20. **;**
         21. **F: '('E')' {$$ = $2; }**
         22. **| NUM {$$ = $1;}**
         23. **;**
         24. **%%**
         25. **int main(int argc, char\* argv[]) {**
         26. **return yyparse();**
         27. **}**
         28. **void yyerror(const char \*s) {**
         29. **printf("%s\n", s);**
         30. **}**
6. 实验结果
   1. 输入与输出1：
   2. 输入与输出2：
   3. 运算正确，能够进行加减乘除并识别出浮点数，并且能正确处理括号、运算符的优先级。