个人信息

学号：1911410

姓名：付文轩

专业：信息安全

学院：网络空间安全学院

作业1、2

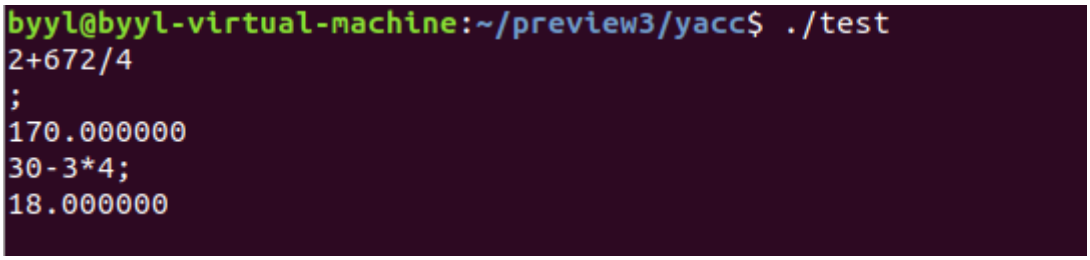
> 由于2仅是1的一个加强版，所以这里将1和2放在一起实现

相关代码为：

```
%{
#include<stdio.h>
#include<stdlib.h>
#ifndef YYSTYPE
#define YYSTYPE double
#endif
int yylex();
extern int yyparse();
FILE* yyin;
void yyerror(const char* s);
%}

%token NUM
%token ADD
%token SUB
%token MUL
%token DEV
%token LB
%token RB

%left ADD SUB
%left MUL DEV
%right UMINUS

%%

lines   :   lines expr ';' { printf("%f\n", $2);}
    |   lines ';'
    |
    ;

 expr   :   expr ADD expr { $$ = $1 + $3; }
    |   expr SUB expr { $$ = $1 - $3; }
    |   expr MUL expr { $$ = $1 * $3; }
    |   expr DEV expr { $$ = $1 / $3; }
    |   LB expr RB { $$ = $2; }
    |   '-' expr %prec UMINUS { $$ = -$2; }
```

```
    |   NUM
    ;


%%

// programs section
int isdigit(int t){

if(t=='0'||t=='1'||t=='2'||t=='3'||t=='4'||t=='5'||t=='6'||t=='7'||t=='8'||t
=='9')return 1;
    else return 0;
}

int yylex()
{
    int temp;
    while(1){
        temp = getchar();
        if(temp==' '||temp=='\t'||temp=='\n'){
            //nothing to do
        }
        else{
        if(isdigit(temp)==1){
            yylval=0;
            while(isdigit(temp)==1){
                yylval=yylval*10+temp-'0';
                temp=getchar();
            }
            ungetc(temp, stdin);
            return NUM;
        }
        if(temp=='+')return ADD;
        if(temp=='-')return SUB;
        if(temp=='*')return MUL;
        if(temp=='/')return DEV;
        if(temp=='(')return LB;
        if(temp==')')return RB;
        return temp;
        }
    }
}

int main()
{
    yyin = stdin;
    do{
        yyparse();
    }while(!feof(yyin));
    return 0;
}

void yyerror(const char* s){
    fprintf(stderr, "Parse error: %s\n", s);
    exit(1);
}
```

实验结果：



作业3

相关代码如下：

```
%{
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#ifndef YYSTYPE
#define YYSTYPE char*
#endif
char idStr[50];
char numStr[50];
int yylex();
extern int yyparse();
FILE* yyin;
void yyerror(const char* s);
%}

%token NUM
%token ID
%token ADD
%token SUB
%token MUL
%token DEV
%token LB
%token RB

%left ADD SUB
%left MUL DEV
%right UMINUS

%%

lines   :   lines expr ';' { printf("%s\n", $2);}
    |   lines ';'
    |
    ;

 expr   :   expr ADD expr { $$ = (char *)malloc(50*sizeof(char));strcpy($$,$1);strcat($$,$3);strcat($$,"+ "); }
    |   expr SUB expr { $$ = (char *)malloc(50*sizeof(char));strcpy($$,$1);strcat($$,$3);strcat($$,"- "); }
    |   expr MUL expr { $$ = (char *)malloc(50*sizeof(char));strcpy($$,$1);strcat($$,$3);strcat($$,"* "); }
```

```
    |   expr DEV expr { $$ = (char *)malloc(50*sizeof(char));strcpy($$,$1);strcat($$,$3);strcat($$,"/ "); }
    |   UMINUS expr %prec UMINUS { $$ = (char *)malloc(50*sizeof(char));strcpy($$,"-"); strcat($$,$2); }
    |   LB expr RB {$$ = (char *)malloc(50*sizeof(char));strcpy($$,$2); }
    |   NUM {$$ = (char *)malloc(50*sizeof(char));strcpy($$,$1);strcat($$," ");}
    |   ID {$$ = (char *)malloc(50*sizeof(char));strcpy($$,$1);strcat($$," ");}
    ;


%%

// programs section
int isdigit(int t){

if(t=='0'||t=='1'||t=='2'||t=='3'||t=='4'||t=='5'||t=='6'||t=='7'||t=='8'||t=='9')return 1;
    else return 0;
}

int isletter(int t){
    if((t>='a' && t<='z') || (t>='A' && t<='Z'))return 1;
    else return 0;
}


int isUMINUS = 1;
int yylex()
{
    int temp;
    while(1){
        temp = getchar();
        if(temp==' '||temp=='\t'||temp=='\n'){
            //nothing to do
        }
        else if(isdigit(temp)==1){
            int indexTemp=0;
            while(isdigit(temp)==1){
                numStr[indexTemp]=temp;
                temp=getchar();
                indexTemp++;
            }
            numStr[indexTemp]='\0';
            yylval=numStr;
            if(temp == '-')
                isUMINUS = 0;
            ungetc(temp, stdin);
            return NUM;
        }
        else if(isletter(temp)==1||(temp=='_')){
            int indexTemp=0;
            while(isletter(temp)==1||(temp=='_')||(isdigit(temp)==1)){
                idStr[indexTemp]=temp;
                temp=getchar();
                indexTemp++;
            }
```

```
            idStr[indexTemp]='\0';
            yylval=idStr;
            if(temp == '-')
                isUMINUS = 0;
            ungetc(temp, stdin);
            return ID;
        }
        else if(temp=='+')return ADD;
        else if(temp=='-'){
            if(isUMINUS==1){
                isUMINUS = 1;
                return UMINUS;
            }
            else {
                isUMINUS = 1;
                return SUB;
            }
        }
        else if(temp=='*')return MUL;
        else if(temp=='/')return DEV;
        else if(temp=='(')return LB;
        else if(temp==')')return RB;
        else return temp;
    }
}

int main()
{
    yyin = stdin;
    do{
        yyparse();
    }while(!feof(yyin));
    return 0;
}

void yyerror(const char* s){
    fprintf(stderr, "Parse error: %s\n", s);
    exit(1);
}
```

实验结果：

```
byyl@byyl-virtual-machine:~/preview3/middlw2behind$ ./mtest
2+A_a2b*(1234-_haha);
2 A_a2b 1234 _haha - * +
-5+2;
-5 2 +
5--2;
5 -2 -
```

思考题

相关代码如下：

```
%{
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct symbol {
    char* name;
    double value;
};
struct symbol sample[10];

int isUMINUS = 1;
int index_of_symbol = 0;

#ifndef YYSTYPE
#define YYSTYPE struct symbol
#endif


char idStr[50];
int yylex();
extern int yyparse();
FILE* yyin;
void yyerror(const char* s);
%}


%token NUM
%token NAME
%token ADD
%token SUB
%token MUL
%token DEV
%token LB
%token RB
%token EQUAL

%left ADD SUB
%left MUL DEV
%right UMINUS

%%

lines   :   lines expr ';' { printf("%f\n", $2.value);}
    |   lines ';'
    |
    ;

 expr   :   expr ADD expr { $$.value = $1.value + $3.value; }
    |   expr SUB expr { $$.value = $1.value - $3.value; }
    |   expr MUL expr { $$.value = $1.value * $3.value; }
    |   expr DEV expr { $$.value = $1.value / $3.value; }
    |   LB expr RB { $$ = $2; }
    |   UMINUS expr %prec UMINUS { $$.name = $2.name; $$.value = -$2.value;
}
    |   NUM { $$ = $1; }
```

```
    |   NAME { $$ = $1; }
    |   NAME EQUAL expr { sample[index_of_symbol].name = (char*)malloc(50*sizeof(char));strcpy(sample[index_of_symbol].name, $1.name);sample[index_of_symbol].value = $3.value;index_of_symbol+=1;$$.value = sample[index_of_symbol].value; }
    ;


%%

// programs section
int isdigit(int t){
    if(t>='0' && t<='9')return 1;
    else return 0;
}

int isletter(int t){
    if((t>='a' && t<='z') || (t>='A' && t<='Z'))return 1;
    else return 0;
}

int yylex()
{
    int temp;
    while(1){
        temp = getchar();
        if(temp==' '||temp=='\t'||temp=='\n'){
            //nothing to do
        }
        else if(isdigit(temp)==1){
            yylval.value = 0;
            yylval.name = "";
            while(isdigit(temp)==1){
                yylval.value = yylval.value*10 + temp - '0';
                temp=getchar();
            }
            isUMINUS = 0;
            ungetc(temp, stdin);
            return NUM;
        }
        else if(isletter(temp)==1||(temp=='_')){
            int indexTemp = 0;
            while(isletter(temp)==1||(temp=='_')||(isdigit(temp)==1)){
                idStr[indexTemp]=temp;
                temp=getchar();
                indexTemp++;
            }
            idStr[indexTemp]='\0';
            yylval.name = idStr;
            for(int i=0;i<index_of_symbol;i++){
                if(!strcmp(sample[i].name, yylval.name)) {
                    yylval.value = sample[i].value;
                    break;
                }
            }
            isUMINUS = 0;
            ungetc(temp, stdin);
            return NAME;
```

```
        }
        else if(temp=='+')return ADD;
        else if(temp=='-'){
            if(isUMINUS == 1) return UMINUS;
            else {
                isUMINUS = 1;
                return SUB;
            }
        }
        else if(temp=='*')return MUL;
        else if(temp=='/')return DEV;
        else if(temp=='(')return LB;
        else if(temp==')')return RB;
        else if(temp=='=')return EQUAL;
        else return temp;
    }
}

int main()
{
    yyin = stdin;
    do{
        yyparse();
    }while(!feof(yyin));
    return 0;
}

void yyerror(const char* s){
    fprintf(stderr, "Parse error: %s\n", s);
    exit(1);
}
```

实验结果：

```
byyl@byyl-virtual-machine:~/preview3/symbol$ ./symbol_test
jack = 4;
0.000000
bob = 2;
0.000000
bob;
2.000000
jack;
4.000000
```