編譯器第二次作業結報

yacc-Yet Another Compiler Compiler

班級: 資訊三丁

學號: D1053020

姓名: 徐葆驊

- 作法:

  Scanner 的設計是讀到 1 個到多個的數字就傳給 parser 最長的那個數字，並且先用 atoi 轉成 int 的形式傳給 parser。其他的符號就是讀到一個就傳一個符號的 token 給 parser。因為輸入的東西有做限制，所以沒有寫額外的 error 的情況出來。

  Parser 的設計就是將高優先權的規則寫在下面，從最上面的規則開始往下展開，這樣高優先權的規則就會先被執行，並且以左結合的方式執行。在最後在判斷是不是有減號加數字的情況將該情況視為負數。

  加分題的做法就是先將中序的算式轉換成後序，這樣就解決了優先順序的問題，然後算出後序的算式就是答案。

- 程式碼:

Scanner

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "calc.tab.h"
%}
%option noyywrap
%%
[0-9]+ { yylval = atoi(yytext);
         return(DIGIT);}
"(" {return(LPAREN);}
")" {return(RPAREN);}
"+" {return(PLUS);}
"-" {return(MINUS);}
"*" {return(MULT);}
"/" {return(DIV);}
\n {return('\n');}

%%
int main()
{
    yyparse();
    return 0;
}
int yyerror(char *msg){
    printf("ERROR: %s\n",msg);
```

```
}
```

Parser

```
%token DIGIT
%token LPAREN
%token RPAREN
%token PLUS
%token MINUS
%token MULT
%token DIV

%left PLUS
%left MINUS
%left MULT
%left DIV
%start list

%%
list :
        | list expr
        {
                printf("ANS is: %d\n\n",$2);
        }
        | list expr '\n'
        {
                printf("ANS is: %d\n\n",$2);
        }
        ;
expr : factor PLUS factor
        {
                $$ = $1 + $3;
                printf("%d plus %d equals %d\n",$1,$3,$$);
        }
        | factor MINUS factor
        {
                $$ = $1 - $3;
               printf("%d MINUS %d equals %d\n",$1,$3,$$);
        }
        | expr PLUS factor
        {
                $$ = $1 + $3;
               printf("%d plus %d equals %d\n",$1,$3,$$);
        }
```

```
41.         | expr MINUS factor
42.         {
43.                 $$ = $1 - $3;
44.                 printf("%d MINUS %d equals %d\n",$1,$3,$$);
45.         }
46.         | factor
47.         {
48.                 $$ = $1;
49.         }
50.         ;
51.  factor : primary MULT primary
52.         {
53.                 $$ = $1 * $3;
54.                 printf("%d MULT %d equals %d\n",$1,$3,$$);
55.         }
56.         | primary DIV primary
57.         {
58.                 $$ = $1 / $3;
59.                 printf("%d DIV %d equals %d\n",$1,$3,$$);
60.         }
61.         | factor MULT primary
62.         {
63.                 $$ = $1 * $3;
64.                 printf("%d MULT %d equals %d\n",$1,$3,$$);
65.         }
66.         | factor DIV primary
67.         {
68.                 $$ = $1 / $3;
69.                 printf("%d DIV %d equals %d\n",$1,$3,$$);
70.         }
71.         | primary
72.         {
73.                 $$ = $1;
74.         }
75.         ;
76.  primary : LPAREN expr RPAREN
77.         {
78.                 $$ = $2;
79.         }
80.         | DIGIT
81.         {
82.                 $$ = $1;
83.         }
84.         | MINUS DIGIT
85.         {
86.                 $$ = -$2;
```

```
87.         }
88. %%
```

加分題

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #include <string.h>
4.
5.  #define MAX 80
6.  #define INTSIZE 10
7.
8.  int num1 = 0;
9.  int num2 = 0;
10. int inputFlag = 0;//0 :num1 還沒有輸入 1 :可以做運算了
11. int numBuffer[MAX];
12.
13. void inToPostfix(char*, char*); //  中序轉後序
14. int priority(char); //  運算子優先權
15. void output(char*); //  輸出結果
16.
17. void inToPostfix(char* infix, char* postfix) {
18.         char stack[MAX] = {'\0'};
19.         int i, j, k,top;// i: infix index, j: postfix index, k: numBuffer index, top: stack top
20.         int numPointer = 0;
21.         char intBuffer[INTSIZE] ; //  整數暫存區
22.         //init
23.         memset(stack, '\0', strlen(stack));
24.         memset(intBuffer, '0', strlen(intBuffer));
25.         for(int i=0;i<MAX;i++){
26.                 numBuffer[i] = 0;
27.         }
28.
29.         for(i = 0, j = 0,k =0 ,top = 0; infix[i] != '\0'; i++) {
30.                 //printf("infix[%d]: %c\n", i, infix[i]);
31.                 switch(infix[i]) {
32.                         case '(':                       //  運算子堆疊
33.                                 stack[++top] = infix[i];
34.                                 break;
35.                         case '+': case '-': case '*': case '/':
36.                                 if(infix[i]=='-'&& i==0){//第一個為負號
37.                                         intBuffer[numPointer++] = infix[i];
38.                                         intBuffer[numPointer] = '\0';//字串結尾加上'\0
39.                                 }else{
```

```
                        if(atoi(intBuffer) != 0){
                        numBuffer[k++] = atoi(intBuffer);//將整數暫存區的字串轉成整數存入 numBuffer
                        memset(intBuffer, '0', strlen(intBuffer));//清空整數暫存區
                        numPointer = 0;//整數暫存區指標歸零
                        postfix[j++] = 'n';//數字 flag
                        }
                        while(priority(stack[top]) >= priority(infix[i])) {
                            postfix[j++] = stack[top--];
                        }
                        stack[++top] = infix[i]; //  存入堆疊
                    }
                    break;
                case ')':
                    numBuffer[k++] = atoi(intBuffer);//將整數暫存區的字串轉成整數存入 numBuffer
                    memset(intBuffer, '0', strlen(intBuffer));//清空整數暫存區
                    numPointer = 0;//整數暫存區指標歸零
                    postfix[j++] = 'n';//數字 flag
                    while(stack[top] != '(') { //  遇 ) 輸出至 (
                        postfix[j++] = stack[top--];
                    }
                    top--;   //  不輸出 (
                    break;
                default:   //  運算元先暫存至整數暫存區
                    intBuffer[numPointer++] = infix[i];
                    intBuffer[numPointer] = '\0';//字串結尾加上'\0
                    //printf("intBuffer: %s\n", intBuffer);
                    break;
            }
        }
        if(atoi(intBuffer) != 0){
            numBuffer[k++] = atoi(intBuffer);//將整數暫存區的字串轉成整數存入 numBuffer
            memset(intBuffer, '0', strlen(intBuffer));//清空整數暫存區
            numPointer = 0;//整數暫存區指標歸零
            postfix[j++] = 'n';//數字 flag
        }
        while(top > 0) { //將最後的結果輸出
            postfix[j++] = stack[top--];
        }
        //for(int tmp = 0; tmp < k; tmp++)printf("numBuffer: %d\n", numBuffer[tmp]);
}

int priority(char op) {
        switch(op) {
```

```
83.             case '+': case '-': return 1;
84.             case '*': case '/': return 2;
85.             default:                    return 0;
86.         }
87. }
88.
89. void output(char* postfix){
90.         int top = -1;
91.         int result = 0;
92.         for(int i = 0; i < strlen(postfix); i++){
93.             switch(postfix[i]){
94.                     case 'n':
95.                             top++;
96.                             break;
97.                     case '+':
98.                             num2 = numBuffer[top];
99.                             num1 = numBuffer[--top];
100.                            result = num1 + num2;
101.                            printf("%d plus %d equals %d\n", num1, num2, result);
102.                            numBuffer[top] = result;
103.                            //將 numBuffer 後面的數字往前移
104.                            for(int j=top+1;j<MAX;j++){
105.                                    numBuffer[j] = numBuffer[j+1];
106.                            }
107.                            break;
108.                    case '-':
109.                            num2 = numBuffer[top];
110.                            num1 = numBuffer[--top];
111.                            result = num1 - num2;
112.                            printf("%d minus %d equals %d\n", num1, num2, result);
113.                            numBuffer[top] = result;
114.                            //將 numBuffer 後面的數字往前移
115.                            for(int j=top+1;j<MAX;j++){
116.                                    numBuffer[j] = numBuffer[j+1];
117.                            }
118.                            break;
119.                    case '*':
120.                            num2 = numBuffer[top];
121.                            num1 = numBuffer[--top];
122.                            result = num1 * num2;
123.                            printf("%d multiply %d equals %d\n", num1, num2, result);
124.                            numBuffer[top] = result;
125.                            //將 numBuffer 後面的數字往前移
126.                            for(int j=top+1;j<MAX;j++){
127.                                    numBuffer[j] = numBuffer[j+1];
128.                            }
```

```
                        break;
                case '/':
                        num2 = numBuffer[top];
                        num1 = numBuffer[--top];
                        result = num1 / num2;
                        printf("%d div %d equals %d\n", num1, num2, result);
                        numBuffer[top] = result;
                        //將 numBuffer 後面的數字往前移
                        for(int j=top+1;j<MAX;j++){
                                numBuffer[j] = numBuffer[j+1];
                        }
                        break;
                default:
                        break;
            }
    }
    printf("ANS is %d\n", result);
}


int main() {
    FILE *fp;
    char infix[MAX];
    char postfix[MAX];
    memset(infix, '\0', strlen(infix));
    memset(postfix, '\0', strlen(postfix));
    for(int i=0;i<MAX;i++){
            numBuffer[i] = 0;
    }

    if (( fp = fopen("input.txt", "r")) == NULL)
    {
            printf("Error! opening file");
            // 文件指针返回 NULL 则退出
            exit(1);
    }
    while(!feof(fp)){
            memset(infix, '\0', strlen(infix));
            memset(postfix, '\0', strlen(postfix));
            fscanf(fp ,"%s", infix);
            printf("infix: %s\n", infix);
            inToPostfix(infix, postfix);
            //printf("postfix: %s\n", postfix);
            output(postfix);
            printf("\n");
    }
```

```
175.
176.        fclose(fp);
177.        return 0;
178.}
```

- 執行結果:

測試資料:

15+32+35-10

65+35

5-3

20-5+5

2*2

2*2*3

6/3

8/2/2

30/2*10+5+6*6

10/2*3+5*3+6

5+10*2

112-(2+32)*3

36/(30+3*2)

12+(12-(12+12))

10*(5+5)

(5+(5+6)*10)-(10/(5-3)+8)

-15*10+200

基本題:

```
15 plus 32 equals 47
47 plus 35 equals 82
82 MINUS 10 equals 72
ANS is: 72

65 plus 35 equals 100
ANS is: 100

5 MINUS 3 equals 2
ANS is: 2

20 MINUS 5 equals 15
15 plus 5 equals 20
ANS is: 20

2 MULT 2 equals 4
ANS is: 4

2 MULT 2 equals 4
4 MULT 3 equals 12
ANS is: 12

6 DIV 3 equals 2
ANS is: 2

8 DIV 2 equals 4
4 DIV 2 equals 2
ANS is: 2

30 DIV 2 equals 15
15 MULT 10 equals 150
150 plus 5 equals 155
6 MULT 6 equals 36
155 plus 36 equals 191
ANS is: 191

10 DIV 2 equals 5
5 MULT 3 equals 15
5 MULT 3 equals 15
15 plus 15 equals 30
30 plus 6 equals 36
ANS is: 36

10 MULT 2 equals 20
5 plus 20 equals 25
ANS is: 25
```

```
2 plus 32 equals 34
34 MULT 3 equals 102
112 MINUS 102 equals 10
ANS is: 10

3 MULT 2 equals 6
30 plus 6 equals 36
36 DIV 36 equals 1
ANS is: 1

12 plus 12 equals 24
12 MINUS 24 equals -12
12 plus -12 equals 0
ANS is: 0

5 plus 5 equals 10
10 MULT 10 equals 100
ANS is: 100

5 plus 6 equals 11
11 MULT 10 equals 110
5 plus 110 equals 115
5 MINUS 3 equals 2
10 DIV 2 equals 5
5 plus 8 equals 13
115 MINUS 13 equals 102
ANS is: 102

-15 MULT 10 equals -150
-150 plus 200 equals 50
ANS is: 50
```

加分題:

```
infix: 15+32+35-10
15 plus 32 equals 47
47 plus 35 equals 82
82 minus 10 equals 72
ANS is 72

infix: 65+35
65 plus 35 equals 100
ANS is 100

infix: 5-3
5 minus 3 equals 2
ANS is 2

infix: 20-5+5
20 minus 5 equals 15
15 plus 5 equals 20
ANS is 20

infix: 2*2
2 multiply 2 equals 4
ANS is 4

infix: 2*2*3
2 multiply 2 equals 4
4 multiply 3 equals 12
ANS is 12

infix: 6/3
6 div 3 equals 2
ANS is 2

infix: 8/2/2
8 div 2 equals 4
4 div 2 equals 2
ANS is 2
```

```
infix: (5+(5+6)*10)-(10/(5-3)+8)
5 plus 6 equals 11
11 multiply 10 equals 110
5 plus 110 equals 115
5 minus 3 equals 2
10 div 2 equals 5
5 plus 8 equals 13
115 minus 13 equals 102
ANS is 102

infix: -15*10+200
-15 multiply 10 equals -150
-150 plus 200 equals 50
ANS is 50
```

```
infix: 30/2*10+5+6*6
30 div 2 equals 15
15 multiply 10 equals 150
150 plus 5 equals 155
6 multiply 6 equals 36
155 plus 36 equals 191
ANS is 191

infix: 10/2*3+5*3+6
10 div 2 equals 5
5 multiply 3 equals 15
5 multiply 3 equals 15
15 plus 15 equals 30
30 plus 6 equals 36
ANS is 36

infix: 5+10*2
10 multiply 2 equals 20
5 plus 20 equals 25
ANS is 25

infix: 112-(2+32)*3
2 plus 32 equals 34
34 multiply 3 equals 102
112 minus 102 equals 10
ANS is 10

infix: 36/(30+3*2)
3 multiply 2 equals 6
30 plus 6 equals 36
36 div 36 equals 1
ANS is 1

infix: 12+(12-(12+12))
12 plus 12 equals 24
24 minus 0 equals 24
12 plus 24 equals 36
ANS is 36

infix: 10*(5+5)
5 plus 5 equals 10
10 multiply 10 equals 100
ANS is 100
```

- 討論

在一開始基本題的時候不知道怎麼樣設定優先權，但是想到老師在第二章的時候有教過優先權越高的寫在越下面，就讓我有了解決方法。

遇到的第二個困難是要怎麼在算完一條算式的時候把最終結果印出來，後來想到這個 parser 最後才會執行最上面的規則，所以就將最後的 print 放在最上面的 list 那邊就可以順利印出規則了。

加分題方面遇到的困難比較多，根據上一次的作業已經可以寫出一個 scanner 了，但是這個 scanner 不能判斷算式的優先順序，原本想說一直往後讀來判斷優先順序，但是後面就覺得不太可行，所以就想到了之前資料結構學的前序中序後序，我先將中序的算式轉換成後序，這樣就可以解決優先順序的問題了，然後只要直接算後序的結果就是答案。

在加分題遇到的第二個問題就是如何判斷負數，因為我在轉後序的時候就把連續讀到的數字蒐集起來，當遇到別的運算符號再把他們轉換成數字放入後序中，但是負號他前面沒有數字，所以再轉換的時候會出問題，又因為不會有括號裡面有負號的情況，像”(-10)”這樣，也就代表負數一定出現在第一個數字，所以我就額外寫了一個如果遇到”-“且又是第一個的情況，就將此情況列為負數。雖然這樣沒辦法解決其他狀況，但是目前負數會出現的情況是這樣所以只好這樣解決。

遇到的第三個問題就是，在 c 裡面讀檔跟處理字串的時候，有時候會有溢位還是什麼情況，就導致資料出問題，像是我在蒐集數字 token 要一次轉換成數字的時候，明明讀到的字元都沒有問題，但是一轉換完馬上出現亂碼，後來我就將讀到字元後，馬上後面在給一個’\0’，讓他轉換的時候不要出問題，就解決了。

- 心得

我認為，這次的基本題比較簡單，因為老師上課就有講過相關的作法，只是將這些規則的寫法寫成程式碼而已。但是加分題的部分就要思考比較多了，首先是字串的處理，之前資料結構學到的中序後序都只有處理單一個字元，就是像 a+b+c，沒有一次處理多個數字，所以就要想辦法做數字的轉換，在來就是負數的問題，所以我認為這次加分題相較於上一次的比較難。