

Project 2— Generating calculator for infix representations with YACC

Source code

```
1  %{
2  #include<stdio.h>
3  #include<ctype.h>
4  #include<math.h>
5  %}
6
7  %token NUMBER
8  %left '+' '-'
9  %left '*' '/'
10 %left '^'
11
12 %%
13 stmt : exp '\n' { printf("=%d\n",$1); }
14       | stmt exp '\n' { printf("=%d\n",$2); }
15       ;
16 exp : exp '+' exp { $$=$1+$3; }
17       | exp '-' exp { $$=$1-$3; }
18       | exp '*' exp { $$=$1*$3; }
19       | exp '/' exp { $$=$1/$3; }
20       | exp '^' exp { $$=pow($1,$3); }
21       | '-' exp { $$=-1*$2; }
22       | '(' exp ')' { $$=$2; }
23       | NUMBER
24       ;
25 %%
26
27 int main(){
28     return yyparse();
29 }
30
31 int yylex(){
32     int c;
33     while((c=getchar())==' ');
34     if(isdigit(c)){
35         ungetc(c,stdin);
36         scanf("%d",&yylval);
37         return (NUMBER);
38     }
39     return (c);
40 }
41
42 void yyerror(char* s){
43     fprintf(stderr,"%s\n",s);
44 }
```

Running result:

```
PS C:\Users\PumpKin\desktop\courses\编原\projects\proj2> ./lab21
12+5-7
=10
3+4*5
=23
2-3*4^2+2
=-44
PS C:\Users\PumpKin\desktop\courses\编原\projects\proj2> _
```