Practical 5

Compiler Construction

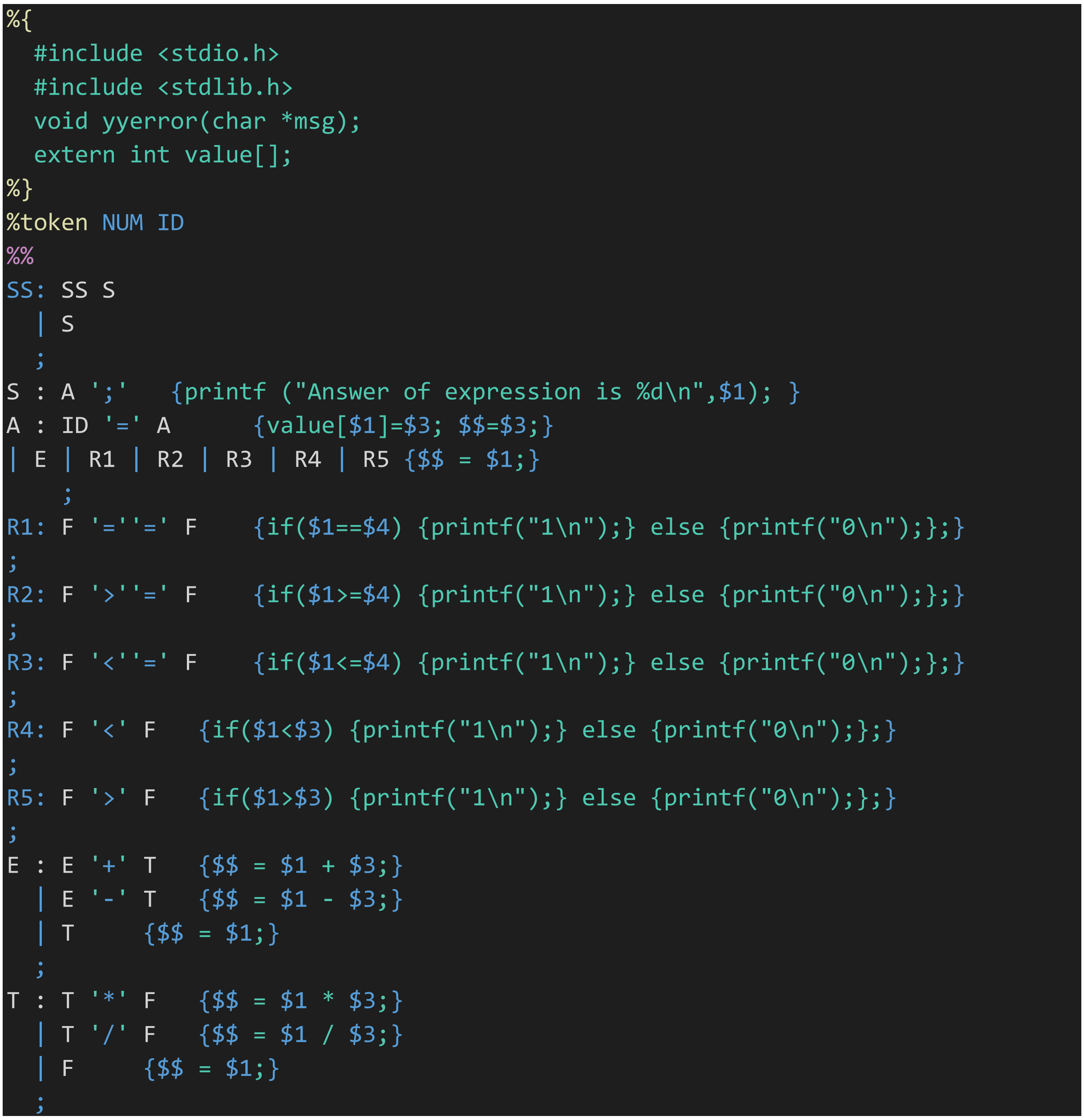
19BCE248

D2

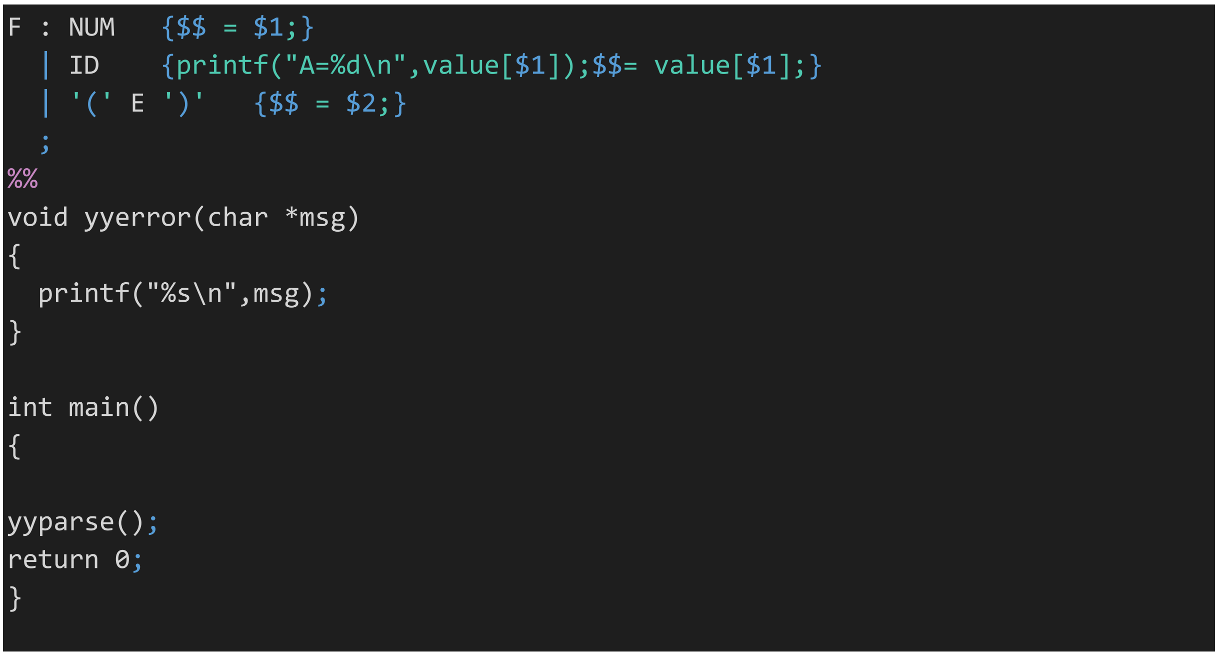
AIM: To implement a calculator in YACC: Syntax Directed Translation.

Code:

Yacc file:



```
%{
  #include <stdio.h>
  #include <stdlib.h>
  void yyerror(char *msg);
  extern int value[];
%}
%token NUM ID
%%
SS: SS S
  | S
  ;
S : A ';'   {printf ("Answer of expression is %d\n",$1); }
A : ID '=' A      {value[$1]=$3; $$=$3;}
| E | R1 | R2 | R3 | R4 | R5 {$$ = $1;}
    ;
R1: F '=''=' F    {if($1==$4) {printf("1\n");} else {printf("0\n");};}
;
R2: F '>''=' F    {if($1>=$4) {printf("1\n");} else {printf("0\n");};}
;
R3: F '<''=' F    {if($1<=$4) {printf("1\n");} else {printf("0\n");};}
;
R4: F '<' F   {if($1<$3) {printf("1\n");} else {printf("0\n");};}
;
R5: F '>' F   {if($1>$3) {printf("1\n");} else {printf("0\n");};}
;
E : E '+' T   {$$ = $1 + $3;}
  | E '-' T   {$$ = $1 - $3;}
  | T     {$$ = $1;}
  ;
T : T '*' F   {$$ = $1 * $3;}
  | T '/' F   {$$ = $1 / $3;}
  | F     {$$ = $1;}
  ;
```

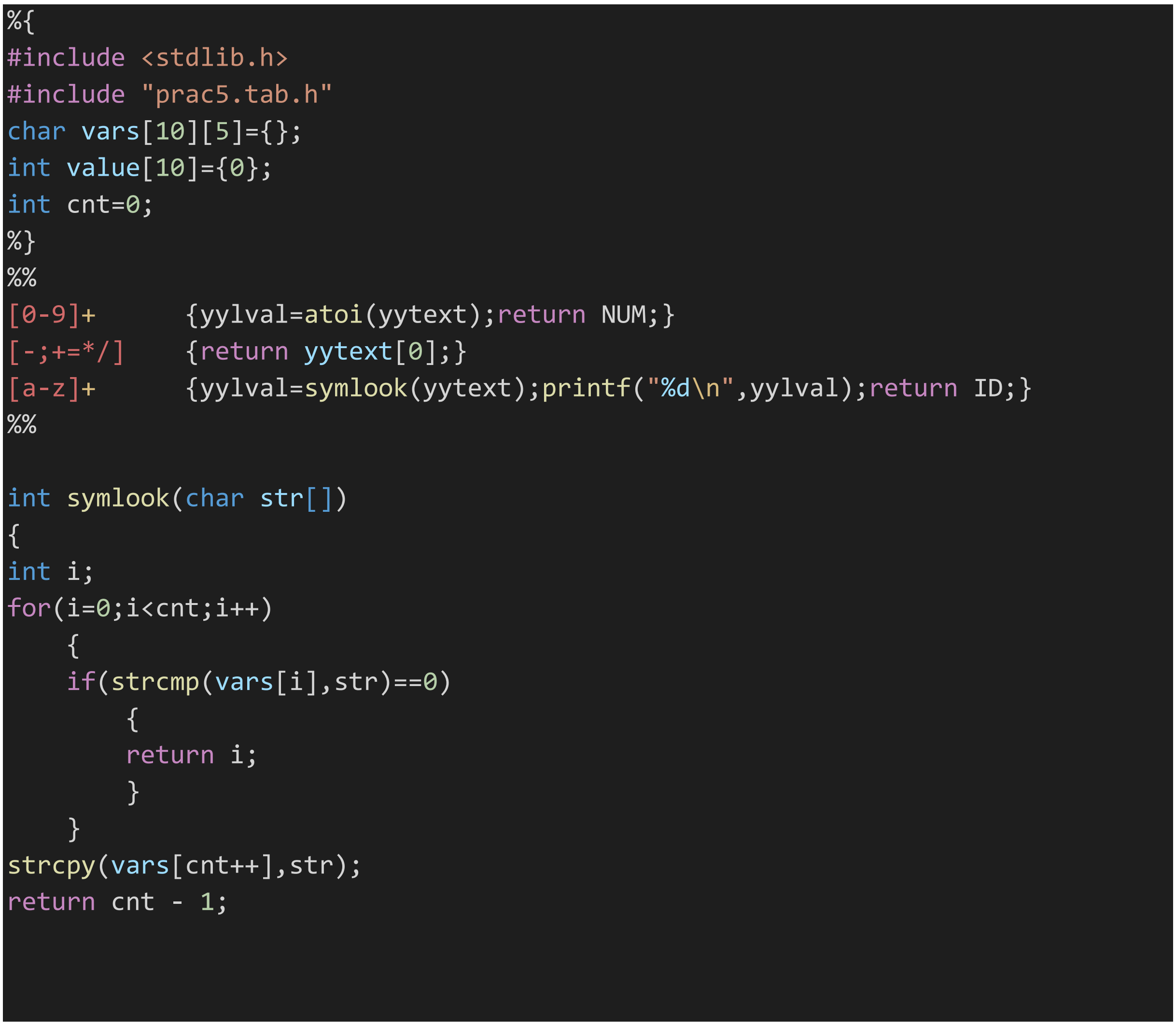


```
F : NUM   {$$ = $1;}
  | ID    {printf("A=%d\n",value[$1]);$$= value[$1];}
  | '(' E ')'   {$$ = $2;}
  ;
%%
void yyerror(char *msg)
{
  printf("%s\n",msg);
}

int main()
{

yyparse();
return 0;
}
```

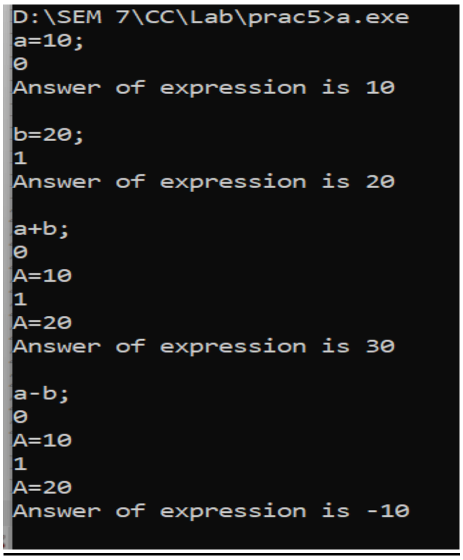
Lex file:



```
%{
#include <stdlib.h>
#include "prac5.tab.h"
char vars[10][5]={};
int value[10]={0};
int cnt=0;
%}
%%
[0-9]+      {yylval=atoi(yytext);return NUM;}
[-;+=*/]    {return yytext[0];}
[a-z]+      {yylval=symlook(yytext);printf("%d\n",yylval);return ID;}
%%

int symlook(char str[])
{
int i;
for(i=0;i<cnt;i++)
    {
    if(strcmp(vars[i],str)==0)
        {
        return i;
        }
    }
strcpy(vars[cnt++],str);
return cnt - 1;
}
int yywrap(){
return 0;
}
```

Output:



```
D:\SEM 7\CC\Lab\prac5>a.exe
a=10;
0
Answer of expression is 10

b=20;
1
Answer of expression is 20

a+b;
0
A=10
1
A=20
Answer of expression is 30

a-b;
0
A=10
1
A=20
Answer of expression is -10
```