1b. Write YACC program to evaluate arithmetic expression involving operators: +, -, *, and /

lex

```
%{
#include "y.tab.h"
extern int yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext); return
NUM;}
[\t]   ;
.   {return yytext[0];}
\n  {return 0;}
%%
```

yacc

```
%{
#include<stdio.h>
#include<stdlib.h>
%}
%token NUM
%left '+' '-'
%left '*' '/'
%%
S:e {printf("result=%d\n",$$); return 0;}
;
e:e'+'e {$$=$1+$3;}
|e'-'e {$$=$1-$3;}
|e'*'e {$$=$1*$3;}
|e'/'e {$$=$1/$3;}
|'('e')' {$$=$2;}
|NUM {$$=$1;}
;
%%
main()
{
printf("enter expr\n");
yyparse();
}
yyerror()
{
printf("error\n");
exit(0);
}
```

NOASSOC

yylval : The yylval global variable is **used to pass the semantic value associated with a token from the lexer to the parser**.