```
%{
  /* Definition section */
#include<stdio.h>
%}

%token NUMBER ID
// setting the precedence
// and associativity of operators
%left '+' '-'
%left '*' '/'

/* Rule Section */
%%
 E : T { printf("Result = %d\n", $$);   return 0;  }

T : T '+' T { $$ = $1 + $3; }
| T '-' T { $$ = $1 - $3; }
| T '*' T { $$ = $1 * $3; }
| T '/' T { $$ = $1 / $3; }
| NUMBER { $$ = $1; }

%%

    int main()
 {
    printf("Enter the expression\n");
    yyparse();
 }

/* For printing error messages */
int yyerror(char* s)
 {
    printf("\nExpression is invalid\n");
 }
```

Save yellow highlighted code with filename.y extension. Example calc.y

LEX Code

```
%{
    /* Definition section*/
#include "y.tab.h"
extern yylval;
%}

%%
[0-9]+  { yylval = atoi(yytext); return NUMBER;  }
[a-zA-Z]+ { return ID; }
[\t]+ ;
\n { return 0; }
```

```
. { return yytext[0]; }
%%
```

Save green highlighted code with filename.y extension. Example calc.l

Note: file name of both lex and yacc should be same, only extension will change .
Example calc.y for yacc file and calc.l for lex file