EXP NO:6

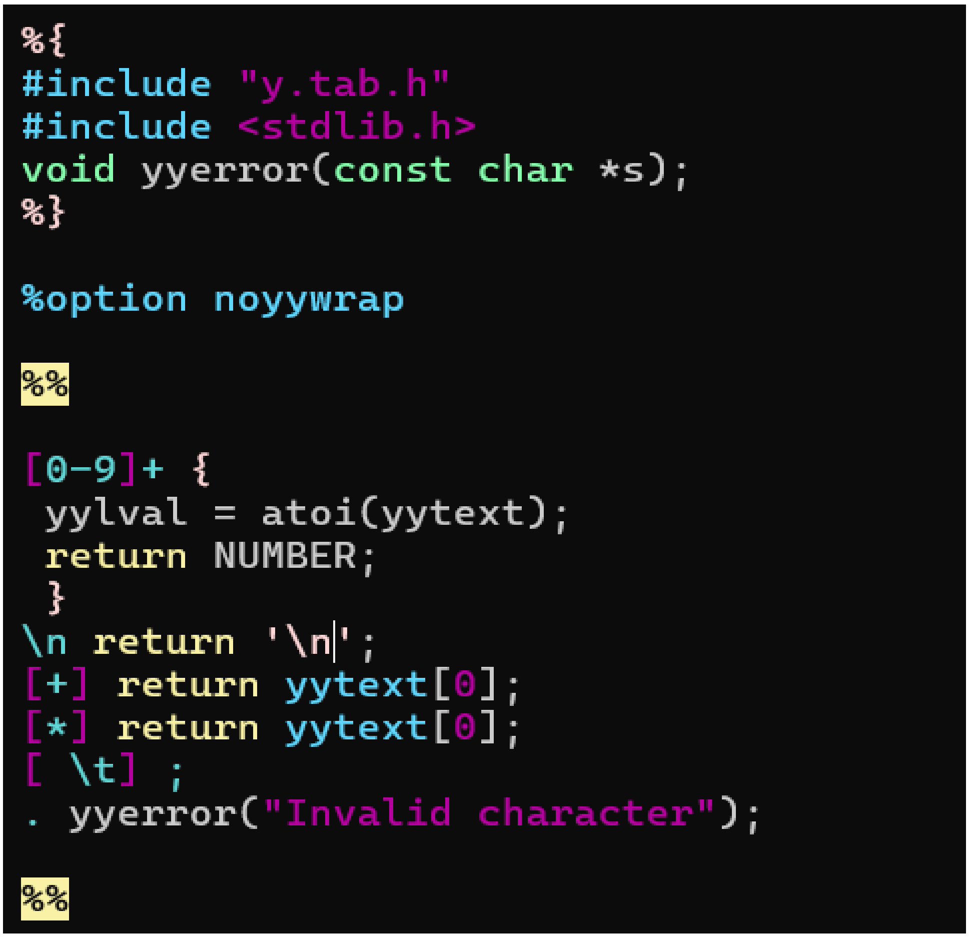
EVALUATE THE EXPRESSION THAT TAKES DIGITS, *, + USING LEX AND YACC

AIM:

To design and implement a LEX and YACC program that evaluates arithmetic expressions containing digits, +, and * while following operator precedence rules.

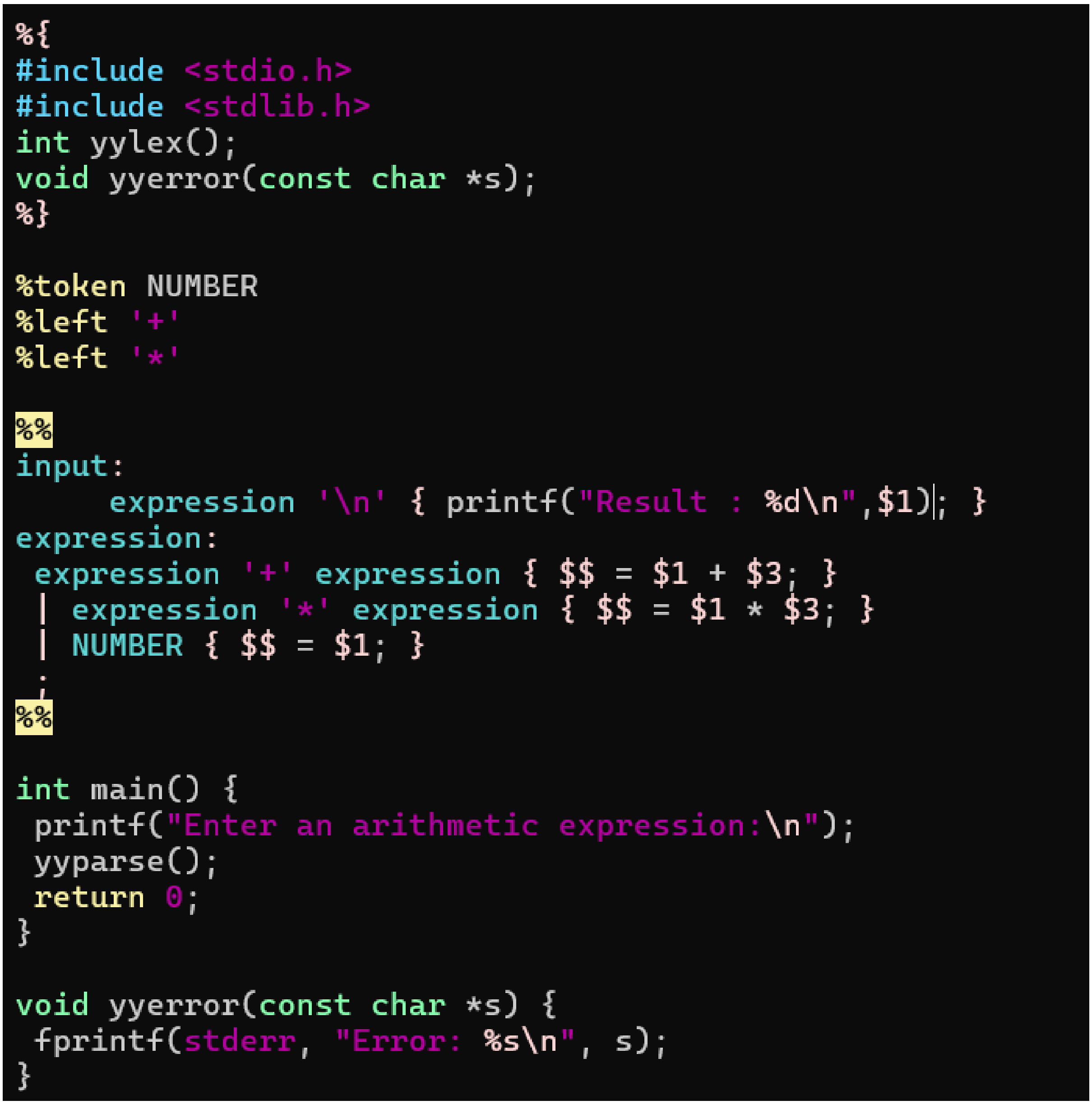
PROGRAM

LEX CODE: ex6.l



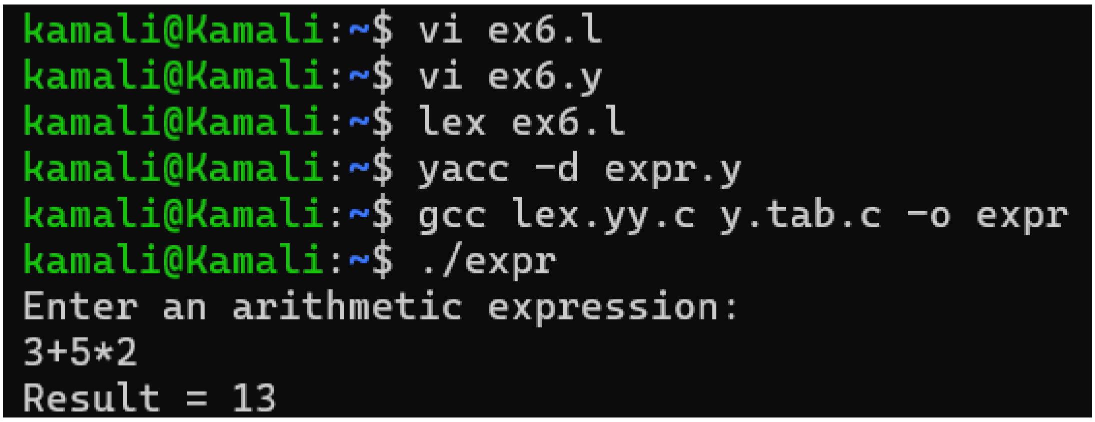
```
%{
#include "y.tab.h"
#include <stdlib.h>
void yyerror(const char *s);
%}

%option noyywrap

%%

[0-9]+ {
 yylval = atoi(yytext);
 return NUMBER;
 }
\n return '\n';
[+] return yytext[0];
[*] return yytext[0];
[ \t] ;
. yyerror("Invalid character");

%%
```

YACC CODE: ex6.y



```
%{
#include <stdio.h>
#include <stdlib.h>
int yylex();
void yyerror(const char *s);
%}

%token NUMBER
%left '+'
%left '*'

%%
input:
     expression '\n' { printf("Result : %d\n",$1); }
expression:
 expression '+' expression { $$ = $1 + $3; }
 | expression '*' expression { $$ = $1 * $3; }
 | NUMBER { $$ = $1; }
 ;
%%

int main() {
 printf("Enter an arithmetic expression:\n");
 yyparse();
 return 0;
}

void yyerror(const char *s) {
 fprintf(stderr, "Error: %s\n", s);
}
```

OUTPUT:



```
kamali@Kamali:~$ vi ex6.l
kamali@Kamali:~$ vi ex6.y
kamali@Kamali:~$ lex ex6.l
kamali@Kamali:~$ yacc -d expr.y
kamali@Kamali:~$ gcc lex.yy.c y.tab.c -o expr
kamali@Kamali:~$ ./expr
Enter an arithmetic expression:
3+5*2
Result = 13
```

RESULT:

Thus the above program to evaluate the expression that takes digits, *, + using lex and yacc is been implemented and executed successfully based on the precedence.