

Aim:

To write the program using LEX and YACC to implement parser on ambiguous grammar.

Algorithm: → File.1

step 1: Start

step 2: Include the necessary header files and declare the necessary variables.

step 3: initialize the digits, operators, parenthesis and return the value else print ~~error~~ Syntax Error

step 4: Call the function & return 1

step 5: Stop

File. y

step 1: Start

step 2: Include the necessary header files and declare the necessary variables

step 3: Substitute the values and calculate respectively for Addition, Subtraction, Multiplication and division and return the result.

step 4: Call the main function and print the result

step 5: Stop.

Program:File 1

```
% option noyywrap
```

```
% {
```

```
    #include <stdio.h>
```

```
    #include "y.tab.h"
```

```
    void yyerror(char *s)
```

```
    extern int yylval;
```

```
% }
```

```
% %:
```

```
[0-9] + {yylval = atoi(yytext);
```

```
return NUM;
```

```
[- + * / \n] {return *yytext;}
```

```
"(" {return *yytext;}
```

```
");" {return *yytext;}
```

```
[\t];
```

```
    {yyerror("Syntax Error");}
```

```
% %:
```

```
int yywrap()
```

```
{
```

```
return 1;
```

```
}
```

File y

```
% {
```

```
    #include <stdio.h>
```

```
    extern int yylex(void);
```

```
    void yyerror(char *);
```

```
% }
```



% token NUM

% %

S:

S expr 'ln' {printf("%d\n", \$2);}

|

;

expr:

expr '+' expr { \$\$ = \$1 + \$3; }

| expr '-' expr { \$\$ = \$1 - \$3; }

| expr '\*' expr { \$\$ = \$1 \* \$3; }

| expr '/' expr { \$\$ = \$1 / \$3; }

| NUM { \$\$ = \$1; }

| '(' expr ')' { \$\$ = \$2; }

;

% %

void yyerror(char \*s)

{

printf("%s\n", s);

}

int main()

{

yy parse();

return 0; }

Result:

Use LEX and YACC to implement parser for ambiguous is executed successfully.

El Mahesh Devaraj\College\Semester\VI Sem\Compiler Design Lab\Lab Exercise\Exercise2. Use LEX and YACC to implement parser for ambiguous grammar\file1 - E

File Edit View Search Document Project Tools Browser ZC Window Help

Directory C:\> C:\ Flex Windows EditPlus Portable App Data Other EditPlus Portable EditPlus Portable

```
1 #include <stdio.h>
2 #include "y.tab.h"
3 void yyerror(char *s);
4 extern int yylval;
5
6
7
8
9
10 [0-9]+ {yylval=atoi(yytext);
11 return NUM;}
12
13 (-+|/|*) {return 'yytext[1];
14 }
15 {return 'yytext[1];
16 }
17 {return 'yytext[1];
18 }
19 {return 'yytext[1];
20 }
21 {return 'yytext[1];
22 }
23 {return 'yytext[1];
24 }
25 {return 'yytext[1];
26 }
27 {return 'yytext[1];
28 }
29 {return 'yytext[1];
30 }
31 {return 'yytext[1];
32 }
33 {return 'yytext[1];
34 }
35 {return 'yytext[1];
36 }
37 {return 'yytext[1];
38 }
39 {return 'yytext[1];
40 }
41 {return 'yytext[1];
42 }
43 {return 'yytext[1];
44 }
45 {return 'yytext[1];
46 }
47 {return 'yytext[1];
48 }
49 {return 'yytext[1];
50 }
51 {return 'yytext[1];
52 }
53 {return 'yytext[1];
54 }
55 {return 'yytext[1];
56 }
57 {return 'yytext[1];
58 }
59 {return 'yytext[1];
60 }
61 {return 'yytext[1];
62 }
63 {return 'yytext[1];
64 }
65 {return 'yytext[1];
66 }
67 {return 'yytext[1];
68 }
69 {return 'yytext[1];
70 }
71 {return 'yytext[1];
72 }
73 {return 'yytext[1];
74 }
75 {return 'yytext[1];
76 }
77 {return 'yytext[1];
78 }
79 {return 'yytext[1];
80 }
81 {return 'yytext[1];
82 }
83 {return 'yytext[1];
84 }
85 {return 'yytext[1];
86 }
87 {return 'yytext[1];
88 }
89 {return 'yytext[1];
90 }
91 {return 'yytext[1];
92 }
93 {return 'yytext[1];
94 }
95 {return 'yytext[1];
96 }
97 {return 'yytext[1];
98 }
99 {return 'yytext[1];
100 }
```

Lex/Yacc Build -----  
Output completed (0 sec consumed) - Normal Termination

All Files (\*)> File Edit View Search Document Project Tools Browser ZC Window Help

for help, press F1

ln 22 col 1 22 00 PC ANSI

El Mahesh Devaraj\College\Semester\VI Sem\Compiler Design Lab\Lab Exercise\Exercise2. Use LEX and YACC to implement parser for ambiguous grammar\file1 - E

File Edit View Search Document Project Tools Browser ZC Window Help

Directory C:\> C:\ Flex Windows EditPlus Portable App Data Other EditPlus Portable EditPlus Portable

```
1 #include <stdio.h>
2 #include "y.tab.h"
3 void yyerror(char *s);
4 extern int yylval;
5
6
7
8
9
10 [0-9]+ {yylval=atoi(yytext);
11 return NUM;}
12
13 (-+|/|*) {return 'yytext[1];
14 }
15 {return 'yytext[1];
16 }
17 {return 'yytext[1];
18 }
19 {return 'yytext[1];
20 }
21 {return 'yytext[1];
22 }
23 {return 'yytext[1];
24 }
25 {return 'yytext[1];
26 }
27 {return 'yytext[1];
28 }
29 {return 'yytext[1];
30 }
31 {return 'yytext[1];
32 }
33 {return 'yytext[1];
34 }
35 {return 'yytext[1];
36 }
37 {return 'yytext[1];
38 }
39 {return 'yytext[1];
40 }
41 {return 'yytext[1];
42 }
43 {return 'yytext[1];
44 }
45 {return 'yytext[1];
46 }
47 {return 'yytext[1];
48 }
49 {return 'yytext[1];
50 }
51 {return 'yytext[1];
52 }
53 {return 'yytext[1];
54 }
55 {return 'yytext[1];
56 }
57 {return 'yytext[1];
58 }
59 {return 'yytext[1];
60 }
61 {return 'yytext[1];
62 }
63 {return 'yytext[1];
64 }
65 {return 'yytext[1];
66 }
67 {return 'yytext[1];
68 }
69 {return 'yytext[1];
70 }
71 {return 'yytext[1];
72 }
73 {return 'yytext[1];
74 }
75 {return 'yytext[1];
76 }
77 {return 'yytext[1];
78 }
79 {return 'yytext[1];
80 }
81 {return 'yytext[1];
82 }
83 {return 'yytext[1];
84 }
85 {return 'yytext[1];
86 }
87 {return 'yytext[1];
88 }
89 {return 'yytext[1];
90 }
91 {return 'yytext[1];
92 }
93 {return 'yytext[1];
94 }
95 {return 'yytext[1];
96 }
97 {return 'yytext[1];
98 }
99 {return 'yytext[1];
100 }
```

Lex/Yacc Build -----  
Output completed (0 sec consumed) - Normal Termination

All Files (\*)> File Edit View Search Document Project Tools Browser ZC Window Help

for help, press F1

ln 4 col 14 31 79 PC ANSI

```
E:\Mohnish Devaraj\College\Semesters\VI Sem\Compiler Design Lab\Lab Exercises\Exercises\2. Use LEX and YACC to implement pars...
4+3
5
4+8
12
12/3
8
```