

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. VIth Sem

Subject Code: CS-316

Subject Name: Compiler Design Lab

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

1. Write a yacc program to evaluate an arithmetic Expression involving +, -, ×, ÷.

→ CODE:

LEX

```
%{
    #include "lab5_1.tab.h"
    extern int yylval;
}%

%%
[0-9]+ {
    yylval = atoi (yytext);
    return NUMBER;
}
[a-zA-Z]+ {return ID;}
[+] {return '+';}
[-] {return '-'};
[*] {return '*'};
[/] { return '/'};
[ \t]+ {;}
\n {return 0;}
. {return yytext[0];}
%%

int yywrap(){
    return 1;
}
```

YACC

```
%{
    #include <stdio.h>
    int flag = 0;
}%

%token NUMBER ID
%left '-' '+'
%right '*' '/'

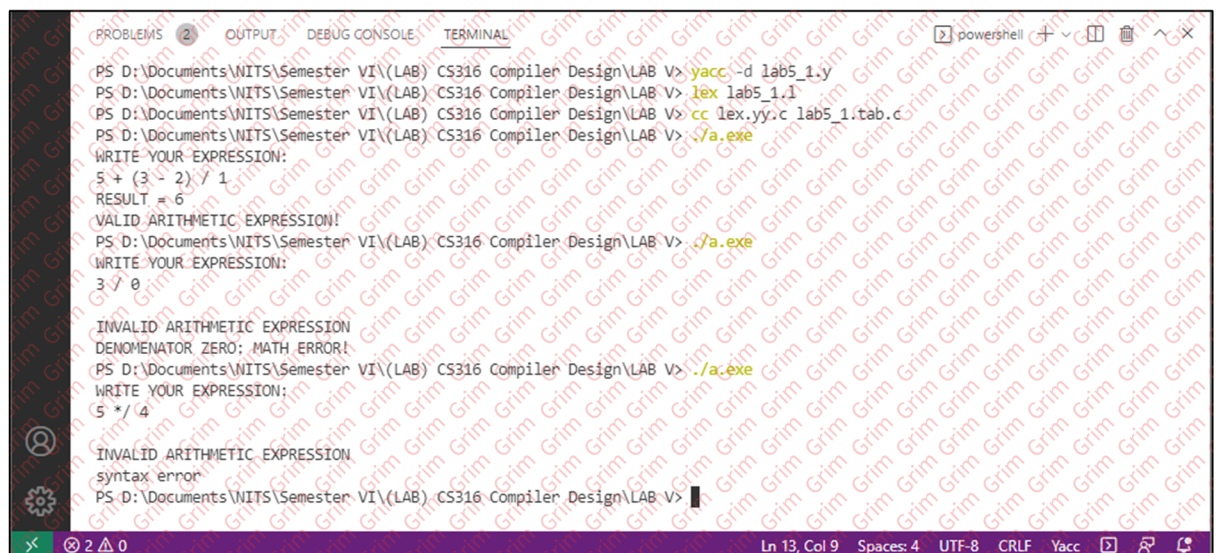
%%
expression: E {
    if (!flag)
        printf ("output: %d\n", $$);
    return 0;
}
E: E '+' E {$$ = $1 + $3;}
| E '-' E {$$ = $1 - $3;}
```

```
|E'*'E {$$ = $1 * $3;}
|E'/'E {$$ = $1 / $3;}
|'-' NUMBER {$$ = -$2;}
|'-' ID {$$ = -$2;}
|'('E')' {$$ = $2;}
|NUMBER {$$ = $1;}
|ID {$$ = $1;};
%%
```

```
int main () {
    printf ("enter your expression:\n");
    yyparse ();
    if (flag == 0)
        printf ("valid!");
    return 0;
}

int yyerror () {
    printf ("\ninvalid\n");
    flag = 1;
    return 0;
}
```

OUTPUT:



```
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V> yacc -d lab5_1.y
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V> lex lab5_1.l
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V> cc lex.yy.c lab5_1.tab.c
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V> ./a.exe
WRITE YOUR EXPRESSION:
5 + (3 - 2) / 1
RESULT = 6
VALID ARITHMETIC EXPRESSION!
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V> ./a.exe
WRITE YOUR EXPRESSION:
3 / 0
INVALID ARITHMETIC EXPRESSION
DENOMINATOR ZERO: MATH ERROR!
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V> ./a.exe
WRITE YOUR EXPRESSION:
5 * / 4
INVALID ARITHMETIC EXPRESSION
syntax error
PS D:\Documents\NITS\Semester VI\LAB\CS316 Compiler Design\LAB V>
```

2. Write a yacc program to recognise nested "IF" control statements, and display the number of levels of listing.

→ CODE:

LEX

```
%{
    #include "lab5_2.tab.h"
}%

%%
"if" {return IF;}
[sS][0-9]* {return S;}
"<"| ">"| "=="| "!="| "<="| ">=" {return RELOP;}
[0-9]+ {return NUMBER;}
[a-zA-Z][a-zA-Z0-9_]* {return ID;}
[ \t]+ {;}
\n {return 0;}
. {return yytext[0];}
%%

int yywrap() {
    return 1;
}
```

YACC

```
%{
    #include <stdio.h>
    #include <stdlib.h>
    int count = 0;
}%

%token IF RELOP S NUMBER ID

%%
statement: if_stat {
    printf ("total nested if statements: %d\n", count);
    return 0;
}
if_stat: IF '(' cond ')' '{' if_stat '}' {count++;}
| S {;}
cond: x RELOP x {;}
x: ID | NUMBER {;};
%%

int main () {
    printf ("write statement:\n");
```

```

    yyparse ();
    return 0;
}

int yyerror (char *str) {
    printf ("invalid!");
    exit (0);
}

```

OUTPUT:

```

PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V> yacc -d lab5_2.y
PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V> lex lab5_2.l
PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V> cc lex.yy.c lab5_2.tab.c
PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V> ./a.exe
ENTER THE STATEMENT:
if (a>b) {if (a>b) {s}}
NUMBER OF NESTED "IF" STATEMENTS = 2
PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V> ./a.exe
ENTER THE STATEMENT:
if (a < b) {s}
NUMBER OF NESTED "IF" STATEMENTS = 1
PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V> ./a.exe
ENTER THE STATEMENT:
if (a == b) {if (a != c) {if (a != d) {s}}
INVALID EXPRESSION!
PS D:\Documents\WITS\Semester VI\LAB) CS316 Compiler Design\LAB V>

```