NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
LP LAB ASSIGNMENT-2

Name: P MANOHAR RAO

Roll No:197158

Section: A

Q1) Write a simple lex/yacc program to show precedence between minus(-) ,plus(+), multiplication(*) operations.

Lex:

```
%{
#include "y.tab.h"
extern yylval;
%}
%%
[0-9]+ {
yylval = atoi(yytext);
return NUMBER;
}
[a-zA-Z]+ { return ID; }
[ \t]+ ; //For skipping whitespaces
\n { return 0; }
. { return yytext[0]; }
%%
```

Yacc;

```
%{
#include <stdio.h>
%}
%token NUMBER ID
// setting the precedence
// and associativity of operators
%left '+' '-'
%left '*' '/'
%%
E : T {
printf("Result = %d\n", $$);
return 0;
}
T :
T '+' T { $$ = $1 + $3; }
| T '-' T { $$ = $1 - $3; }
```

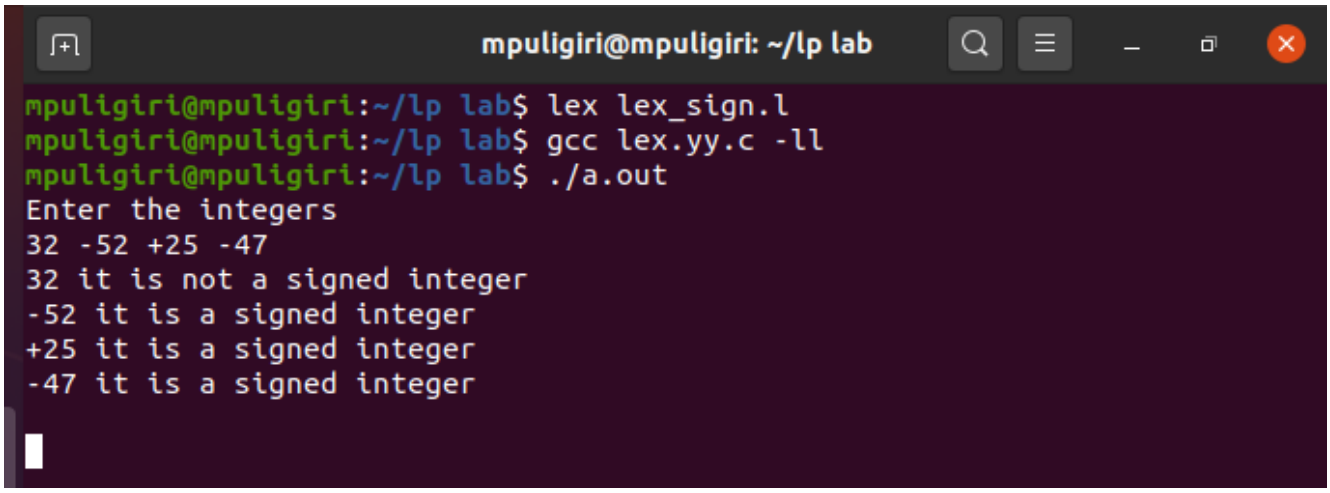
```
| T '*' T { $$ = $1 * $3; }
| T '/' T { $$ = $1 / $3; }
| '-' NUMBER { $$ = -$2; }
| '-' ID { $$ = -$2; }
| '(' T ')' { $$ = $2; }
| NUMBER { $$ = $1; }
| ID { $$ = $1; };
%%
int main() {
printf("Enter the expression\n");
yyparse();
}
int yyerror(char* s) {
printf("\nExpression is invalid\n");}
```

```
mpuligiri@mpuligiri:~/lp lab$ lex
a.out        cnd.l        cnw.l        lex_prec.l   nonalp.l     yacc_pre
c.y
cls.l        cnvc.l       ctn.l        lex.yy.c     set3to5.l
mpuligiri@mpuligiri:~/lp lab$ lex lex_prec.l
mpuligiri@mpuligiri:~/lp lab$ yacc -d yacc_prec.y -ll
mpuligiri@mpuligiri:~/lp lab$ cc lex.yy.c y.tab.c -ll
lex_prec.l:3:8: warning: type defaults to 'int' in declaration of 'yylval
' [-Wimplicit-int]
    3 | extern yylval;
      |        ^~~~~~
y.tab.c: In function 'yyparse':
y.tab.c:1220:16: warning: implicit declaration of function 'yylex' [-Wimp
licit-function-declaration]
 1220 |       yychar = yylex ();
      |                ^~~~~
y.tab.c:1389:7: warning: implicit declaration of function 'yyerror'; did
you mean 'yyerrok'? [-Wimplicit-function-declaration]
 1389 |       yyerror (YY_("syntax error"));
      |       ^~~~~~~
      |       yyerrok
mpuligiri@mpuligiri:~/lp lab$ ./a.out
Enter the expression
8*7+2-6
Result = 52
mpuligiri@mpuligiri:~/lp lab$
```

Q2) Write a simple lex program to find out whether any entered number is signed integer or not.

```
%{
#include<stdio.h>
%}
```

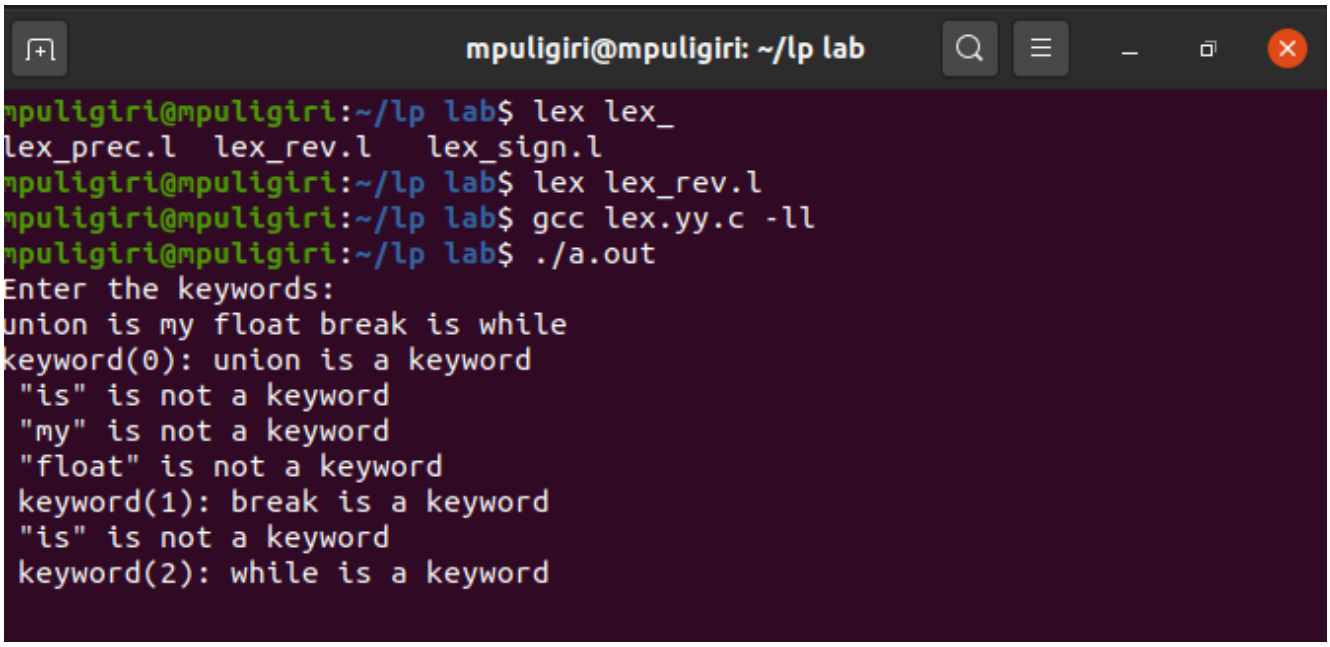
```
%%
[+|-][0-9]+ {printf("%s it is a signed integer\n",yytext);}
[0-9]+ { printf("%s it is not a signed integer\n",yytext);}
[ \t]+ ; //For skipping whitespaces
%%
int main(){
printf("Enter the integers\n");
yylex();
return 0;
}
```



Q3) Write a lex program that could recognise few reserved words in C language

```
%{
int count;
//program to recognize the keywords
%}
%%
[%\t]+ {}
auto |
double |
if |
static |
break |
else | int |
struct |
case |
enum | long |
switch | char | extern |
near |
typedef | const | float | register |
union | unsigned | void |
while |
default {printf("keyword(%d): %s is a keyword\n",count++,yytext);}
[a-zA-Z]+ {printf("\"%s\" is not a keyword\n", yytext);}
%%
```

```
int main()
{
printf("Enter the keywords:\n");
yylex();
}
```



Q4) Write lex and yacc program to do the processing of E+E|E-E|(E)| id where id is a number.

Lexx:

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

Yacc:

```
%{
#include<stdio.h>
%}
%token NUMBER
// setting the precedence
%left '+' '-'
%%
E : T { printf("Result = %d\n", $1); return 0; }
T :
T '+' T { $$ = $1 + $3; }
| T '-' T { $$ = $1 - $3; }
| '(' T ')' { $$ = $2; }
```

```
| NUMBER { $$ = $1; }
%%
int main() {
printf("Enter the expression\n");
yyparse();
}
int yyerror(char* s) {
printf("\nExpression is invalid\n");
}
```

```
mpuligiri@mpuligiri: ~/lp lab
mpuligiri@mpuligiri:~/lp lab$ lex lex_eprocess.l
mpuligiri@mpuligiri:~/lp lab$ yacc -d yacc_eprocess.y
mpuligiri@mpuligiri:~/lp lab$ cc lex.yy.c y.tab.c -ll
lex_eprocess.l:3:8: warning: type defaults to 'int' in declaration of 'yy
lval' [-Wimplicit-int]
    3 | extern yylval;
      |        ^~~~~~
y.tab.c: In function 'yyparse':
y.tab.c:1214:16: warning: implicit declaration of function 'yylex' [-Wimp
licit-function-declaration]
 1214 |       yychar = yylex ();
      |                ^~~~~
y.tab.c:1371:7: warning: implicit declaration of function 'yyerror'; did
you mean 'yyerrok'? [-Wimplicit-function-declaration]
 1371 |       yyerror (YY_("syntax error"));
      |       ^~~~~~~
      |       yyerrok
mpuligiri@mpuligiri:~/lp lab$ ./a.out
Enter the expression
3+5-8+3
Result = 3
mpuligiri@mpuligiri:~/lp lab$
```