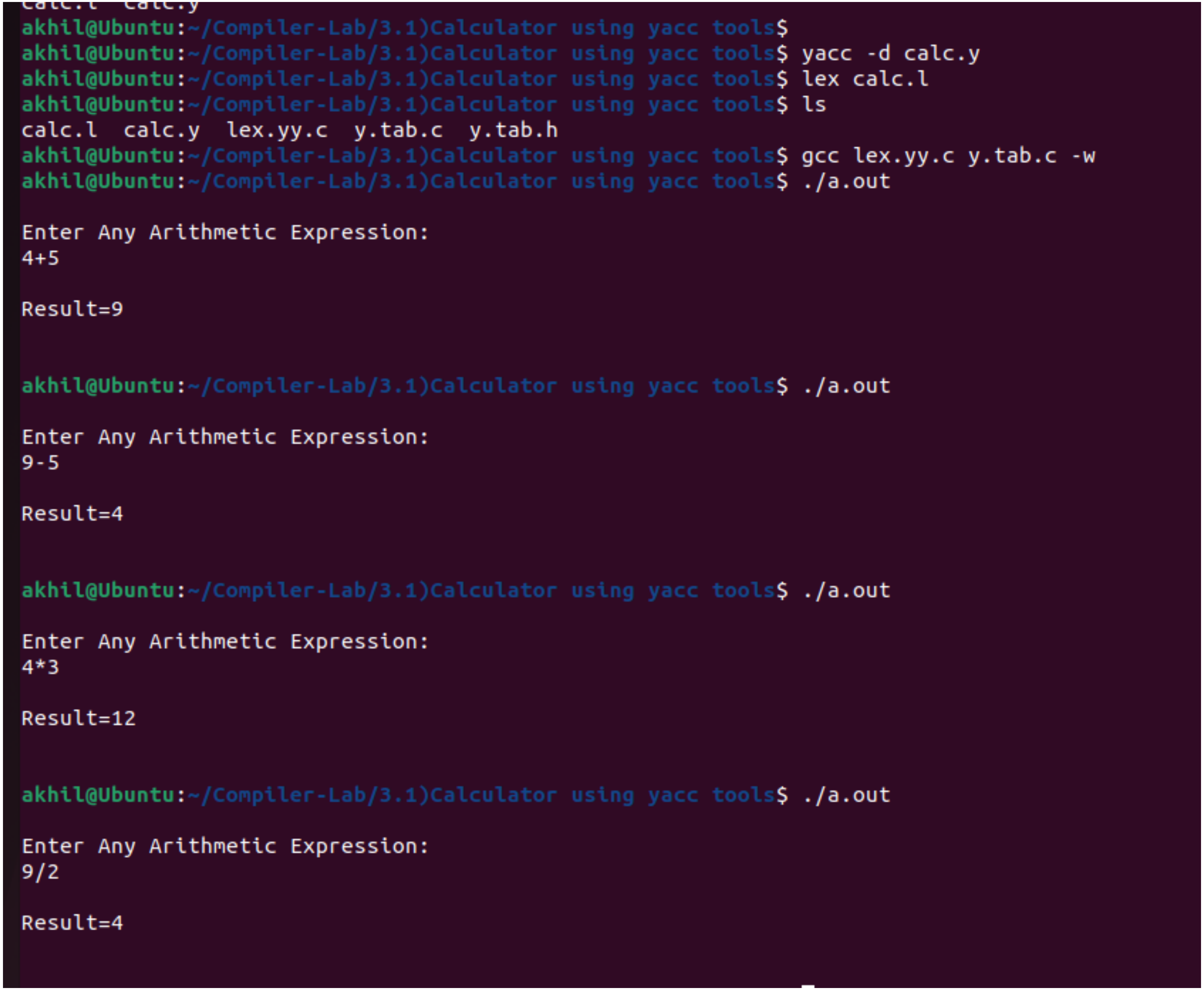
```
%{
/* Definition section */
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
%}

/* Rule Section */
%%
[0-9]+ {
                yylval=atoi(yytext);
                return NUMBER;

        }
[\t] ;

[\n] return 0;

. return yytext[0];

%%

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

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

%token NUMBER
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'
%%
ArithmeticExpression: E{
         printf("\nResult=%d\n",$$);
         return 0;
        };

E:E'+'E {$$=$1+$3;}
 |E'-'E {$$=$1-$3;}
 |E'*'E {$$=$1*$3;}
 |E'/'E {$$=$1/$3;}
 |E'%'E {$$=$1%$3;}
 |'('E')' {$$=$2;}
 | NUMBER {$$=$1;}
;
%%
void main()
{
 printf("\nEnter Any Arithmetic Expression:\n");
 yyparse();
   if(flag==0)
   printf("\n\n");
}
void yyerror()
{
   printf("\nInvalid\n\n");
   flag=1;
}
```



```
calc.l  calc.y
akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$
akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ yacc -d calc.y
akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ lex calc.l
akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ ls
calc.l  calc.y  lex.yy.c  y.tab.c  y.tab.h
akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ gcc lex.yy.c y.tab.c -w
akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ ./a.out

Enter Any Arithmetic Expression:
4+5

Result=9


akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ ./a.out

Enter Any Arithmetic Expression:
9-5

Result=4


akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ ./a.out

Enter Any Arithmetic Expression:
4*3

Result=12


akhil@Ubuntu:~/Compiler-Lab/3.1)Calculator using yacc tools$ ./a.out

Enter Any Arithmetic Expression:
9/2

Result=4
```