Exercise 5 – Implementation of Desk Calculator using Yacc Tool

Harshini S
185001058

Program Code:

lexx.l

```
%{
#include<stdio.h>
#include<stdlib.h>
#include "y.tab.h"
void yyerror(char *);
extern int yylval;
%}

%%
[ \t]+ ;
[0-9]+ {yylval = atoi(yytext);
return INTEGER;}
[-+*/^] {return *yytext;}
"<<" {return LSHIFT;}
">>" {return RSHIFT;}
"&&" {return *yytext;}
"||" {return *yytext;}
"!" {return *yytext;}
"(" { return *yytext;}
")" {return *yytext;}
\n {return *yytext;}
. {char msg[25];
sprintf(msg, "%s<%s>","Invalid Character",yytext);
yyerror(msg);}
%%
```

calc.y

```
%{
  #include<stdio.h>
  #include<stdlib.h>
  #include<math.h>
  #include"y.tab.h"
  int yylex(void);
```

```
  void yyerror(char *);
  int flag=0;

%}

%token INTEGER LSHIFT RSHIFT

%%

prg : line prg
    | line

line : expr '\n' {printf("%d\n",$1);
}
expr : expr LSHIFT expr { $$ = $1 << $3; }
     | expr RSHIFT expr { $$ = $1 >> $3;
     }

expr : expr '||' andex {$$ = $1 || $3;}
     | andex {$$ = $1;}

andex : andex '&&' notex {$$ = $1 && $3;}
      | notex {$$=$1;}

notex : '!' addex {$$ = !$1;}
      | addex {$$=$1;}

addex : addex '+' mulex { $$ = $1 + $3;}
     | addex '-' mulex { $$ = $1 - $3;}
     | mulex {$$ = $1;}

mulex : mulex '*' powex { $$ = $1 * $3;}
      | mulex '/' powex { $$ = $1 /
      $3;} | mulex '%' powex { $$ = $1
      % $3;} | powex { $$ = $1;}
powex : powex '^' term {$$ = pow($1, $3);}
      | term {$$=$1;}

term : '(' expr ')' {$$=$2;}
     | INTEGER {$$=$1;}

%%
```

```
//driver code
int main()
{
   yyparse();
   if(flag==0)
   printf("\nEntered arithmetic expression is Valid\n\n");
   return 0;
}
void yyerror(char *s)
{
   fprintf(stderr,"%s\n",s);
   flag=1;
   return;
}
int yywrap(){
  return 1;
}
```

ip.txt

```
3+9
3+9*6
(3+4)*7
(3-4)+(7*6)
5/7+2
4^2^1
(2^3)^2
```