**3. Use LEX and YACC tool to implement Desktop Calculator.**

File.l

%{

#include<stdlib.h>

#include"y.tab.h"

voidyyerror(char \*s);

externintyylval;

%}

%%

[0-9]+ {yylval=atoi(yytext); return INT;}

[a-z] {yylval=toascii(\*yytext)-97; return ID;}

[A-Z] {yylval=toascii(\*yytext)-65; return ID;}

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

\( {return \*yytext;}

")" {return \*yytext;}

[\t] ;

. {yyerror("Invalid Token!!");}

%%

intyywrap()

{

return 1;

}

File.y

%{

#include<stdio.h>

externintyylex(void);

voidyyerror(char \*);

int x=0;

intval[26];

%}

%token INT ID

%%

nithish:

nithishexpr '\n' {x=$2; printf("%d\n",$2);}

|nithish ID '=' expr '\n' {val[$2]=$4;}

|

;

expr:

expr '+' T {$$=$1+$3;}

|expr '-' T {$$=$1-$3;}

|T {$$=$1;}

|'+' T {$$=x+$2;}

|'-' T {$$=x-$2;}

;

T:

F {$$=$1;}

|T '\*' F {$$=$1\*$3;}

|T '/' F {$$=$1/$3;}

|'\*' F {$$=x\*$2;}

|'/' F {$$=x/$2;}

;

F:

INT {$$=$1;}

|ID {$$=val[$1];}

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

;

%%

voidyyerror(char\* s)

{

printf("%s",s);

}

int main()

{

yyparse();

return 0;

}