**Practical – 8**

**Aim:**

**(a) Write a YACC program to implement simple desk calculator.**

**(b) Write a YACC program to validate the syntax of type declaration of variables.**

**Code (a):**

**practical8a.l**

%{

#include "y.tab.h"

extern int yylval;

%}

%%

[0-9]+ {

/\* convert yytext to an integer \*/

yylval = atoi(yytext);

return NUM;

}

[ \t] /\* Ignore whitespaces \*/;

\n return 0;

. return yytext[0];

%%

**practical8b.y**

%{

#include <stdio.h>

void yyerror(char \*s);

int yylex(void);

%}

%token NUM

%left '+' '-' '\*' '/' '%'

%left UNARY\_MINUS

%%

S: EXPR {printf("Result: %d \n", $$);}

EXPR: EXPR '+' EXPR { $$ = $1 + $3; }

| EXPR '-' EXPR { $$ = $1 - $3; }

| EXPR '\*' EXPR { $$ = $1 \* $3; }

| EXPR '/' EXPR { $$ = $1 / $3; }

| EXPR '%' EXPR { $$ = $1 % $3; }

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

| '-' EXPR %prec UNARY\_MINUS { $$ = -$2; }

| NUM { $$ = $1; }

;

%%

void yyerror(char \*s) {

fprintf(stderr, "%s: detected\n", s);

}

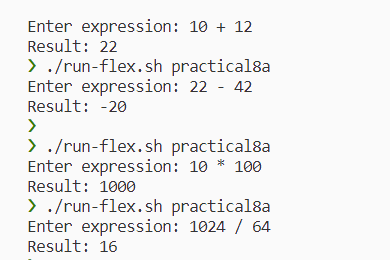
int main() {

printf("Enter expression: ");

return yyparse();

}

**Output (a):**

****

**Code (b):**

**practical8b.l**

%{

#include "y.tab.h"

%}

KEYWORD int|char|float|double|void|struct|union|enum|typedef|extern|static|auto|register|const|volatile|signed|unsigned|short|long|if|else|switch|case|default|while|do|for|goto|continue|break|return|sizeof|asm|inline|restrict

LETTER [a-zA-Z\_]

CONSTANT [0-9]

EQUALS =

SEMICOLON ;

COMMA ,

SPACE [ \t]+

%%

{KEYWORD}\* {

printf("KEYWORD\t=>\t%s\n", yytext);

return KEYWORD;

}

{LETTER}+({LETTER}|{CONSTANT})\* {

printf("ID\t=>\t%s\n", yytext);

return ID;

}

{CONSTANT}+\.?({CONSTANT})\* {

printf("CONST\t=>\t%s\n", yytext);

return CONST;

}

{EQUALS} {

printf("EQUAL\n");

return EQ;

}

{SEMICOLON} {

printf("SEMI\n");

return SEMI;

}

{COMMA} {

printf("COMMA\n");

return COMMA;

}

{SPACE} {

if (yytext[0] == ' ')

printf("SPACE\n");

else

printf("TAB SPACE\n");

return SPACE;

}

\n return NEWLINE;

%%

**practical8b.y**

%{

#include <stdio.h>

void yyerror(char \*s);

int yylex(void);

%}

%token KEYWORD ID CONST EQ SEMI COMMA SPACE NEWLINE

%%

S1: S NEWLINE {

printf("Declaration is valid\n");

return 0;

}

S: KEYWORD SPACE A SEMI

A: ID|ID COMMA A|ID EQ CONST|ID EQ CONST COMMA A|;

%%

void yyerror(char \*s) {

fprintf(stderr, "%s: detected\n", s);

}

int main() {

printf("Enter declaration: ");

return yyparse();

}

**Output (b):**

