Name :- Shlok Zanwar

Class :- TY-C3

Roll No :- 323071

GR No. :- 21910163

======================================================= **Assignment 4**

**Part A**

**A4a.l**

%{

#include<stdio.h>

#include "y.tab.h"

%}

%%

[0-9]+ {yylval=atof(yytext);return NUMBER;}

[\t ] ;

[\n] {return 0;}

. {return yytext[0];}

%%

int yywrap()

{

return 0;

}

**A4a.y**

%{

#include<stdio.h>

#include "y.tab.h"

int yylex();

int yyerror();

%}

%token NUMBER

%%

arithExp: exp {printf("Ans= %d\n",$$);}

exp: exp '+' mulExp

{$$ = $1 + $3;}

| exp '-' mulExp

{$$ = $1 - $3;}

| mulExp

;

mulExp: mulExp '\*' primary

{$$ = $1 \* $3;}

| mulExp '/' primary

{$$ = $1 / $3;}

| primary

;

primary: '(' exp ')' {$$

= $2;}

| '-' primary

{$$ = $2;}

| NUMBER

;

%%

int main()

{

printf("\nEnter Expression: ");

yyparse();

return 1;

}

int yyerror(char \*s)

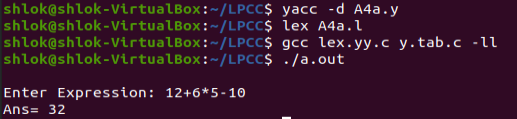
{

printf("%s",s);

return 1;

}

Output:



**Part B**

**A4b.l**

%{

#include<stdio.h>

#include<math.h>

#include <string.h>

#include "y.tab.h"

%}

%%

[ \t]+ ;

sqrt {return SQRT;}

exp {return EXP;}

pow {return POW;}

log {return LOG;}

strlen {return STRLEN;}

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

[a-zA-Z][a-zA-Z0-9\_]+ {yylval = yytext+1;return STRING;}

. {return yytext[0];}

\n { return 0; }

%%

int yywrap()

{

return 0;

}

**A4b.y**

%{

#include<stdio.h>

#include<math.h>

int yylex();

void yyerror();

%}

%token SQRT EXP STRLEN NUMBER STRING LOG POW

%%

stmt : sqroot {printf("=%d",$1);return 0;}

|

expo {printf("=%d",$1);return 0;}

|

power {printf("=%d",$1);return 0;}

|

logs {printf("=%d",$1);return 0;}

|

strLength {printf("=%d",$1);return 0;}

;

sqroot : SQRT '(' NUMBER ')' {$$=sqrt($3);} ;

expo : EXP '(' NUMBER ')' {$$=exp($3);} ;

logs : LOG '(' NUMBER ')' {$$=log($3);} ;

power : POW '(' NUMBER ',' NUMBER ')'

{$$=pow($3,$5);} ;

strLength : STRLEN '(' STRING ')' {$$=strlen($3);} ;

%%

int main()

{

yyparse();

return 1;

}

void yyerror()

{

printf("This is invalid Statement!\n");

}

**Part C**

**A4c.l**

%{

#include<stdio.h>

#include "y.tab.h"

%}

%%

new return NEW;

"[" return OPEN\_SQ;

"]" return CLOSE\_SQ;

"=" return EQ;

"," return COMMA;

"\_" return UD;

(["\t"])+ return WS;

[a-zA-Z]+[a-zA-Z0-9\_]\* return CHAR;

[0-9]+ return DIGIT;

\n return 0;

%%

**A4c.y**

%{

#include<stdio.h>

#include "y.tab.h"

int yylex();

int yyerror();

%}

%token BUILTIN UD WS CHAR OPEN\_SQ CLOSE\_SQ EQ NEW SC

COMMA DIGIT

%%

start : varlist WS varlist {printf(" NOT Valid Declaration \n");}

| varlist UD DIGIT {printf("Valid Declaration \n");}

| varlist {printf("Valid Declaration \n");}

| varlist UD varlist {printf("Valid Declaration \n");}

;

varlist : varlist COMMA CHAR | CHAR

;

%%

int yywrap()

{ return 1;

}

int main()

{

printf("\nEnter variable : ");

yyparse();

return 1;

}

int yyerror(char \*s)

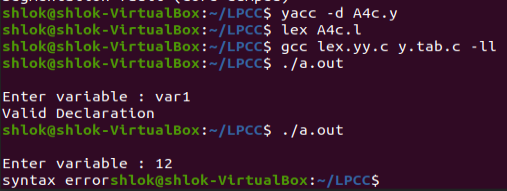
{

printf("%s",s);

return 1;

}

**Output:**

****