**Name : Aditi Nikam Roll No : 323045**

**GR No : 21910513 Class : TY Comp C2**

**4 A**

**Code:**

**Lex Code:**

%{

#include <stdio.h>

#include "y.tab.h"

extern yylval;

%}

%%

[0-9]+ {

yylval = atoi(yytext);

return NUM;

}

[a-zA-Z]+ { return ID; }

[ \t]+ ;

\n { return 0; }

. { return yytext[0]; }

%%

int yywrap()

{

return 1;

}

**Yaac Code:**

%{

#include <stdio.h>

void yyerror();

int yylex();

%}

%token NUM ID

%left '+' '-'

%left '\*' '/'

%%

E : T {

printf("Result = %d\n", $$);

return 0;

}

T :

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

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

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

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

| '-' NUM { $$ = -$2; }

| '-' ID { $$ = $2; }

| '(' T ')' { $$ = $2; }

| NUM { $$ = $1; }

| ID { $$ = $1; };

%%

int main()

{

printf("Enter the expression: ");

yyparse();

}

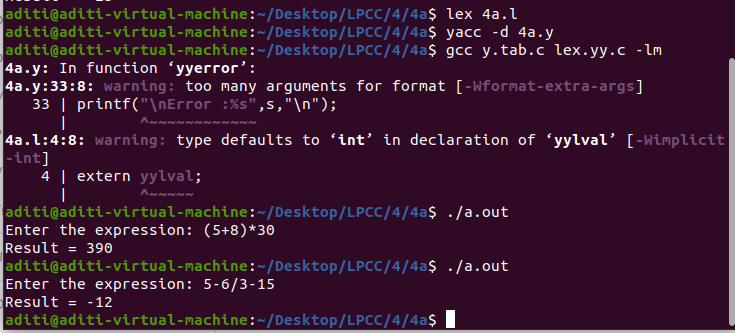
void yyerror(char\* s)

{

printf("\nError :%s",s,"\n");

}

**Output:**



**4 B**

**Code:**

**Lex Code:**

%{

/\* Definition section\*/

#include "y.tab.h"

#include<string.h>

extern yylval;

%}

%%

sqrt { return SQRT;}

strlen {return STRLEN;}

[0-9]+ {

yylval=atoi(yytext);

return NUMBER;

}

' { return t;}

[a-zA-Z]+ { printf("%s\n",yytext); return string; }

[\t]+; /\*For skipping whitespaces\*/

\n { return 0; }

. { return yytext[0];}

%%

**Yaac Code:**

%{

#include <math.h>

#include<string.h>

#include <stdio.h>

%}

%token SQRT NUMBER

%token STRLEN string t

%%

s : sq {printf("Square of number= %d\n", $1); return 0;}

| str{printf("Length of string = %d\n",$1); return 0;}

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

str: STRLEN '('string')' {$$=strlen($3);} ;

%%

int main() {

printf("Enter the expression\n");

//int a = strlen("abcd");

//printf("%d",a);

yyparse();

}

/\* For printing error messages \*/

int yyerror(char\* s) {

printf("\nExpression is invalid\n");

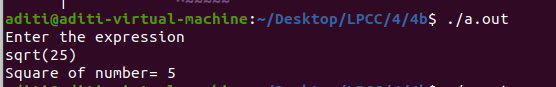
}

int yywrap(){

return 1;

}

**Output:**



**4 C**

**Code:**

**Lex Code:**

%{

#include "y.tab.h"

#include <stdio.h>

#include <stdlib.h>

%}

%%

[a-zA-Z] return Letter;

[0-9] return Number;

"\_" return UnderScore;

\n return 0;

. return yytext[0];

%%

int yywrap()

{

return 1;

}

**Yaac Code:**

%{

#include <stdio.h>

#include <stdlib.h>

int valid = 1;

%}

%token Letter Number UnderScore

%%

S : Letter Number

| S Letter

| S Number

| S UnderScore S

| Letter

;

%%

int yyerror()

{

printf("\nIts not a identifier!\n");

valid=0;

return 0;

}

int main()

{

printf("\nEnter a string to be tested :");

yyparse();

if(valid)

{

printf("\nIt is a identifier!\n");

}

}

**Output:**

