PROGRAM TO RECOGNIZE A VALID ARITHMETIC EXPRESSION

Ex.No:3.a

Date:

AIM:

To write a c program to recognize a valid arithmetic expression.

ALGORITHM:

Step1: Start the program.

Step2: Reading an expression .

Step3: Checking the validating of the given expression according to the rule using yacc.

Step4: Using expression rule print the result of the given values

Step5: Stop the program.

PROGRAM:

LEX Program(Validarith.l)

%{

#include<stdio.h>

#include "Validarith.tab.h"

%}

%%

[a-zA-Z]+ return VARIABLE;

[0-9]+ return NUMBER;

[\t] ;

[\n] return 0;

. return yytext[0];

%%

int yywrap()

{

return 1;

}

YACC Program (Validarith.y)

%{

#include <stdio.h>

#include <stdlib.h>

// Function prototype

int yylex();

void yyerror(const char \*s);

%}

%token NUMBER

%token VARIABLE

%left '+' '-'

%left '\*' '/' '%'

%left '(' ')'

%%

// Grammar rules

S: E {

printf("\nEntered arithmetic expression is valid\n\n");

return 0;

}

;

E: E '+' E

| E '-' E

| E '\*' E

| E '/' E

| E '%' E

| '(' E ')'

| NUMBER

| VARIABLE

;

%%

int main() {

printf("\nEnter any arithmetic expression: \n");

yyparse();

return 0;

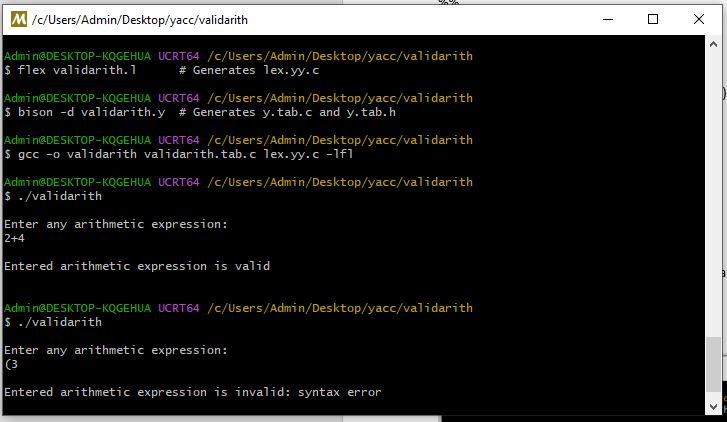
}

void yyerror(const char \*s) {

printf("\nEntered arithmetic expression is invalid: %s\n\n", s);

}

Output:



CONCLUSION:

Thus a program to recognize a valid arithmetic expression was executed successfully.

PROGRAM TO RECOGNIZE A VALID VARIABLE WHICH STARTS WITH A

LETTER FOLLOWED BY ANY NUMBER OF LETTERS OR DIGITS

Ex.No:3.b

Date:

AIM :

To write a yacc program to check valid variable followed by letter or digits

ALGORITHM:

Step1: Start the program

Step2: Reading an expression

Step3: Checking the validating of the given expression according to the rule using yacc.

Step4: Using expression rule print the result of the given values

Step5: Stop the program

PROGRAM CODE:

LEX Program(Valid\_identifier.l)

%{

#include "valid\_identifier.tab.h"

%}

%%

[a-zA-Z][a-zA-Z0-9]\* { return IDENTIFIER; }

. { return 0; } /\* For any other invalid characters \*/

%%

int yywrap() {

return 1;

}

YACC Program(Valid\_identifier.y):

%{

#include <stdio.h>

#include <ctype.h>

// Declaration of yylex function

int yylex(void);

// Declaration of yyerror function

int yyerror(const char \*s);

%}

%token IDENTIFIER

%%

start:

identifier\_check

;

identifier\_check:

IDENTIFIER { printf("It is a valid identifier!\n"); }

| /\* error handling for invalid input \*/

{ printf("It is not a valid identifier!\n"); }

;

%%

int yyerror(const char \*s) {

printf("It is not a valid identifier!\n");

return 0;

}

int main() {

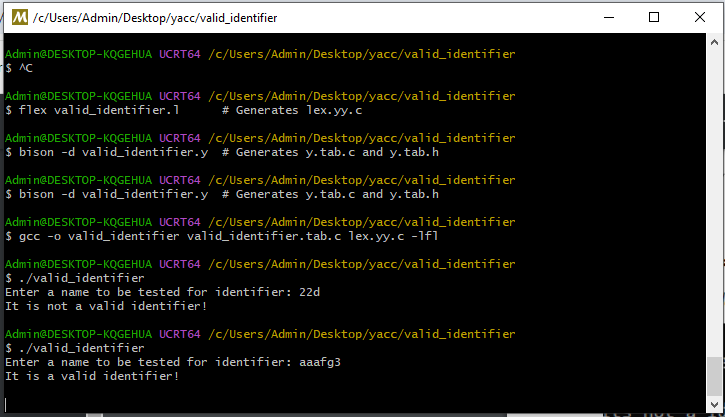
printf("Enter a name to be tested for identifier: ");

yyparse();

return 0;

}

Output:



CONCLUSION:

Thus a program to check valid variable followed by letter or digits was executed successfully.