**ASSIGNMENT NO.**

NAME: ROHIT ANIL GUND

CLASS: BE COMP-1 ROLL NO.: 402055

**PROGRAM:**

//-----------------------------------Lex file----------------------------------//

%{

#include "y.tab.h"

extern int yylval;

%}

%%

[0-9]+ {yylval = atoi(yytext); return NUMBER;} /\* cast pointer to int for compiler warning \*/

[ \t\n]+ ;

"+" return(PLUS);

"-" return(MINUS);

"\*" return(TIMES);

"/" return(DIVIDE);

";" return(END);

%%

int yywrap (void) {return 1;}

//-----------------------------------Yacc file---------------------------------------//

%{

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct node

{

struct node \*left;

struct node \*right;

char \*data;

} node;

node \*mknode(char \*data,node \*left, node \*right);

void display(node \*root);

#define YYSTYPE node \*

%}

%token NUMBER

%token END

%token PLUS MINUS TIMES DIVIDE

%left TIMES

%left PLUS MINUS

%%

stmt : exp END {printf("\nThe Abstract Syntax Tree is (preorder traversal) : "); display($1); printf("\n");}

;

exp : exp1 {$$ = $1;}

| exp PLUS exp1 {$$ = mknode("+",$1,$3);}

| exp MINUS exp1 {$$ = mknode("-",$1,$3);}

;

exp1 : num {$$ = $1;}

| exp1 TIMES num {$$ = mknode("\*",$1,$3);}

| exp1 DIVIDE num {$$ = mknode("/",$1,$3);}

;

num : NUMBER {char buf[10]; snprintf(buf, sizeof(buf), "%d",yylval); $$ = mknode(buf,0,0);}

;

%%

int main (void)

{

printf("Enter an arithmetic expression: \t");

return yyparse();

}

node \*mknode(char \*data,node \*left,node \*right)

{

node \*newnode = (node \*)malloc(sizeof(node));

char \*newstr = (char \*)malloc(strlen(data)+1);

strcpy(newstr,data);

newnode->left = left;

newnode->right = right;

newnode->data = newstr;

return(newnode);

}

void display(node \*root)

{

if (root->left || root->right)

printf("(");

printf(" %s ", root->data);

if (root->left)

display(root->left);

if (root->right)

display(root->right);

if (root->left || root->right)

printf(")");

}

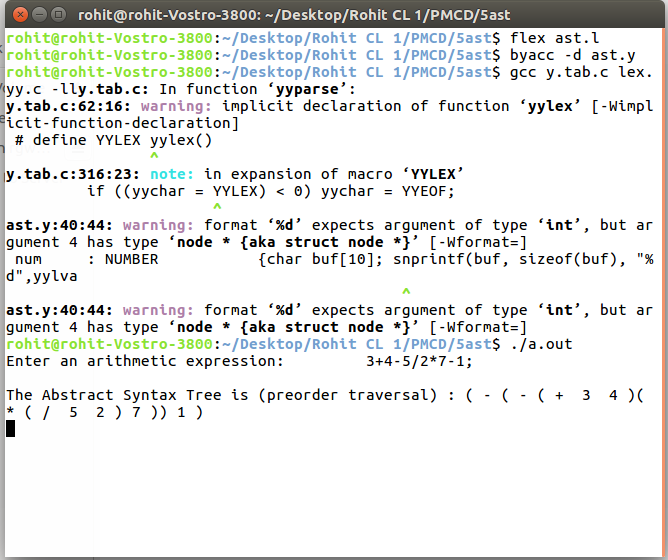
int yyerror (char \*s)

{

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

}

**OUTPUT :**

****