**PRACTICAL-3**

**AIM:**

Write an ambiguous CFG to recognize an infix expression and implement a parser that recognizes the infix expression using YACC.

**IMPLEMENTATION:**

* yacc <filename with .y extension>
* gcc <newly created .c file> -o <file name for exe file>
* <filename of exe file>

**PROGRAM CODE:**

%{

/\*\*\* Auxiliary declarations section \*\*\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

/\* Custom function to print an operator\*/

void print\_operator(char op);

/\* Variable to keep track of the position of the number in the input \*/

int pos=0;

char p;

%}

/\*\*\* YACC Declarations section \*\*\*/

%token NUM

%left '+'

%left '\*'

%%

/\*\*\* Rules Section \*\*\*/

start : expr '\n' {exit(1);}

;

expr: expr '+' expr {print\_operator('+');}

| expr '\*' expr {print\_operator('\*');}

| '(' expr ')'

| NUM {printf("%c ",p);}

;

%%

/\*\*\* Auxiliary functions section \*\*\*/

void print\_operator(char c){

switch(c){

case '+' : printf("+ ");

break;

case '\*' : printf("\* ");

break;

}

return;

}

yyerror(char const \*s)

{

printf("yyerror %s",s);

}

yylex(){

char c;

c = getchar();

p=c;

if(isdigit(c)){

pos++;

return NUM;

}

else if(c == ' '){

yylex(); /\*This is to ignore whitespaces in the input\*/

}

else {

return c;

}

}

main()

{

printf("\nPARTH PATEL\n19DCS098\n");

yyparse();

return 1;

}

**OUTPUT:**



