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:

```
PS C:\00_SEM_7\3_CS450_DESIGN_OF_LANGUAGE_PROCESSORS\1_PRACTICALS\0_PRE_BUILT\Pract-3-Infix> yacc in.y
PS C:\00_SEM_7\3_CS450_DESIGN_OF_LANGUAGE_PROCESSORS\1_PRACTICALS\0_PRE_BUILT\Pract-3-Infix> gcc infix.tab.c -o program
infix.tab.c: In function 'yyparse':
infix.tab.c:589:16: warning: implicit declaration of function 'yylex' [-Wimplicit-function-declaration]
 # define YYLEX yylex ()
                ^
infix.tab.c:1249:16: note: in expansion of macro 'YYLEX'
       yychar = YYLEX;
                ^~~~
infix.tab.c:1403:7: warning: implicit declaration of function 'yyerror' [-Wimplicit-function-declaration]
       yyerror (YY_("syntax error"));
       ^~~~~~~
infix.y: At top level:
infix.y:47:1: warning: return type defaults to 'int' [-Wimplicit-int]
 yyerror(char const *s)
 ^~~~~~~
infix.y:52:1: warning: return type defaults to 'int' [-Wimplicit-int]
 yylex(){
 ^~~~~
infix.y: In function 'yylex':
infix.y:56:8: warning: implicit declaration of function 'isdigit' [-Wimplicit-function-declaration]
     if(isdigit(c)){
        ^~~~~~~
infix.y: At top level:
infix.y:68:1: warning: return type defaults to 'int' [-Wimplicit-int]
 main()
 ^~~~
```

```
PS C:\00_SEM_7\3_CS450_DESIGN_OF_LANGUAGE_PROCESSORS\1_PRACTICALS\0_PRE_BUILT\Pract-3-Infix> ./program
1+(2*3)+1
1 2 3 * + 1 +
```