4a) Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.

```
%{
#include <stdio.h>
int ext(int);
int a[]={0,0,0,0},valid=1,opnd=0,top=-1,i;
%}

%x oper

%%
["("]       {top++;}
[")"]       {top--;}
[a-zA-Z0-9]+      {BEGIN oper; opnd++;}
<oper>"+"                 {if(valid) {valid = 0; i = 0;} else ext(0);}
<oper>"-"                 {if(valid) {valid = 0; i = 1;} else ext(0);}
<oper>"*"                 {if(valid) {valid = 0; i = 2;} else ext(0);}
<oper>"/"                 {if(valid) {valid = 0; i = 3;} else ext(0);}
<oper>"("                 {top++;}
<oper>")"                 {top--;}
<oper>[a-zA-Z0-9]+        {opnd++; if(valid == 0) {valid = 1; a[i]++;} else ext(1);}
<oper>"\n"                {if(valid == 1 && top == -1) {printf("Valid expression\n"); return 0;} else
                           ext(2);}
.\n                       ext(4);
%%

int yywrap() {  }

int ext(int x)
{
        printf("\nInvalid expression%d\n",x);
        exit(0);
```

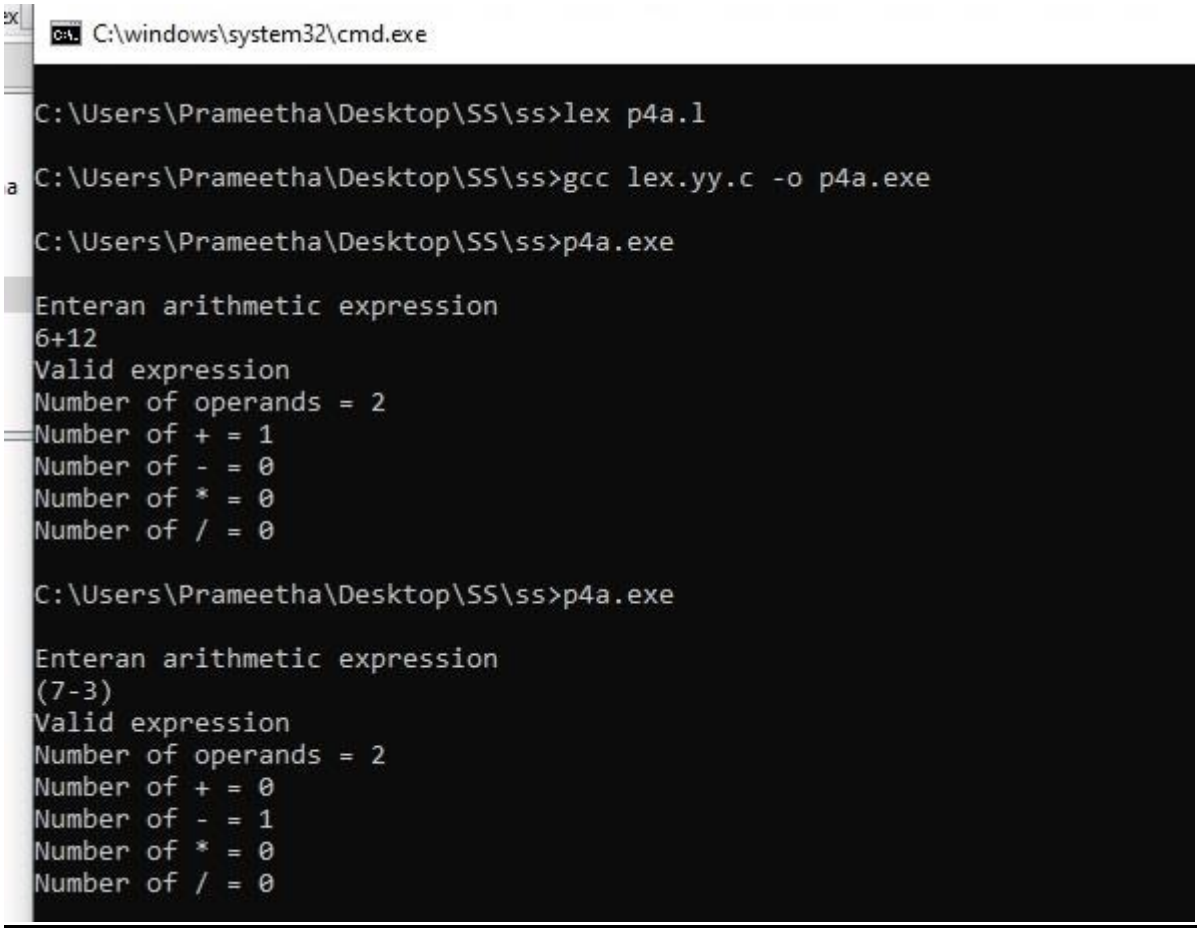
```
}

int main()
{
        printf("\nEnteran arithmetic expression\n");
        yylex();
        printf("Number of operands = %d\n",opnd);
        printf("Number of + = %d\n",a[0]);
        printf("Number of - = %d\n",a[1]);
        printf("Number of * = %d\n",a[2]);
        printf("Number of / = %d\n",a[3]);
        return 0;
}
```

Commands For execution

```
lex pgm_name.l
gcc lex.yy.c -o pgm_name.exe
pgm_name.exe
```

Output


C:\windows\system32\cmd.exe
C:\Users\Prameetha\Desktop\SS\ss>lex p4a.l
C:\Users\Prameetha\Desktop\SS\ss>gcc lex.yy.c -o p4a.exe
C:\Users\Prameetha\Desktop\SS\ss>p4a.exe
Enteran arithmetic expression
6+12
Valid expression
Number of operands = 2
Number of + = 1
Number of - = 0
Number of * = 0
Number of / = 0
C:\Users\Prameetha\Desktop\SS\ss>p4a.exe
Enteran arithmetic expression
(7-3)
Valid expression
Number of operands = 2
Number of + = 0
Number of - = 1
Number of * = 0
Number of / = 0