

1a. Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately

```
% {
int a[]={0,0,0,0},i,valid=1,opnd=0;
% }

%x OPER

%%
[^a-z ^A-Z ^0-9]+ ext();
[a-z A-Z 0-9]+ { BEGIN OPER;opnd++;}
<OPER>"+" {if(valid) { valid=0;i=0;} else ext();}
<OPER>"-" {if(valid) { valid=0;i=1;} else ext();}
<OPER>"*" {if(valid) { valid=0;i=2;} else ext();}
<OPER>"/" {if(valid) { valid=0;i=3;} else ext();}
<OPER>[a-z A-Z 0-9]+ { opnd++;if(valid==0) { valid=1;a[i]++;}else ext();}
<OPER>"\n" {if(valid==0) ext();else return 0;}
.\n ext();
%%

int main()
{
printf("enter expr\n");
yylex();
printf("valid\n");
printf("operands=%d\n",opnd);
printf("+=%d\n",a[0]);
printf("-=%d\n",a[1]);
printf("*=%d\n",a[2]);
printf("/=%d\n",a[3]);
}
int ext()
{
printf("invalid\n");
exit(0);
}
```

To run the Program,

gedit filename.l

lex filename.l

cc lex.yy.c -ll

./a.out