1.

```
%{
#include<stdio.h>
int lines=0,chars=0,words=0,spaces=0;
%}
%%
\n        {lines++;}
\t        {spaces+=4;}
[ ]       {spaces++;}
[^ \t\n#]+ {words++;chars+=yyleng;} //[^ \t\n#] not space, tab, line, #
# {return 0;}
%%
int yywrap()
{
        return 1;
}
int main()
{
        printf("Enter the string:\n");
        yylex();
        printf("Lines: %d\nWords: %d\nChars: %d\nSpaces: %d\n",lines,words,chars,spaces);
        return 0;
}
```

```
flex program1a.l
gcc lex.yy.c -o program1a -lfl
./program1a
```

(1b) lex file

```
%{
#include "y.tab.h"
%}
%%
"a"      {return 'a';}
"b"      {return 'b';}
"c" {return 'c';}
.        {return yytext[0];}
\n       {return 0;}
%%
```

(yacc file)

```
%{
#include<stdio.h>
#include<stdlib.h>
int yyerror();
int yylex();
%}
%%
S:A B
;
A:'a'A'b'
|
;
B:'b'B'c'
|
;
%%
int main()
{
        printf("Enter the input:\n");
        yyparse();
        printf("Valid string\n");
}
int yyerror()
{
        printf("Invalid string\n");
        exit(0);

}
```

```
yacc -d pg1b.y
flex pg1b.l
gcc y.tab.c lex.yy.c -o output -lfl
./output
```

(2a): lex file

```
%{
#include <stdio.h>
int posint = 0, negint = 0, posfrac = 0, negfrac = 0;
%}

num [0-9]+

posint \+?{num}
negint -{num}

posnum \+?{num}\/\+?{num}|-{num}\/-{num}
negnum -{num}\/\+?{num}|\+?{num}\/-{num}

%%

{posint} posint++;

{negint} negint++;

{posnum} posfrac++;

{negnum} negfrac++;

[ \t] ;

[\n] return 0;

. ECHO;

%%

int yywrap(){}

int main() {
        yylex();
        printf("Positive integers: %d\n", posint);
        printf("Negative integers: %d\n", negint);
        printf("Positive fractions: %d\n", posfrac);
        printf("Negative fractions: %d\n", negfrac);
}
```