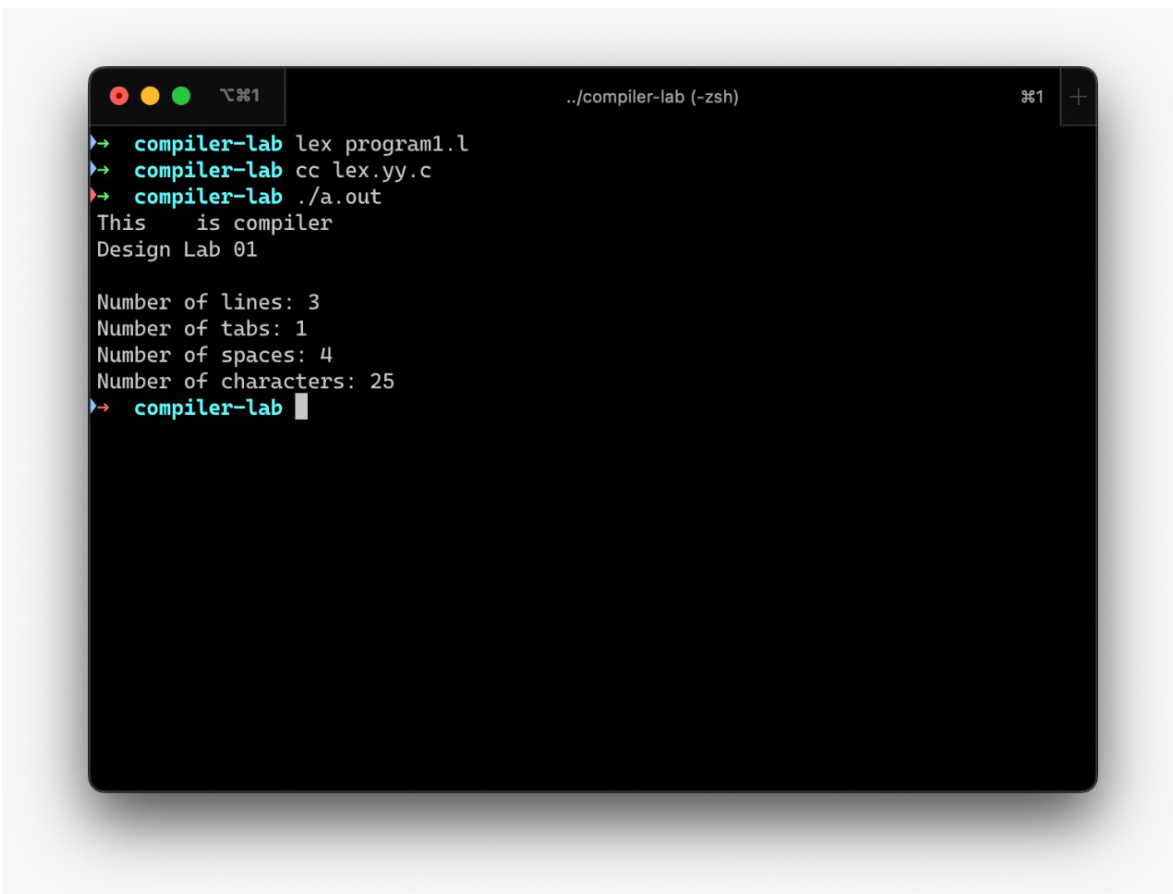
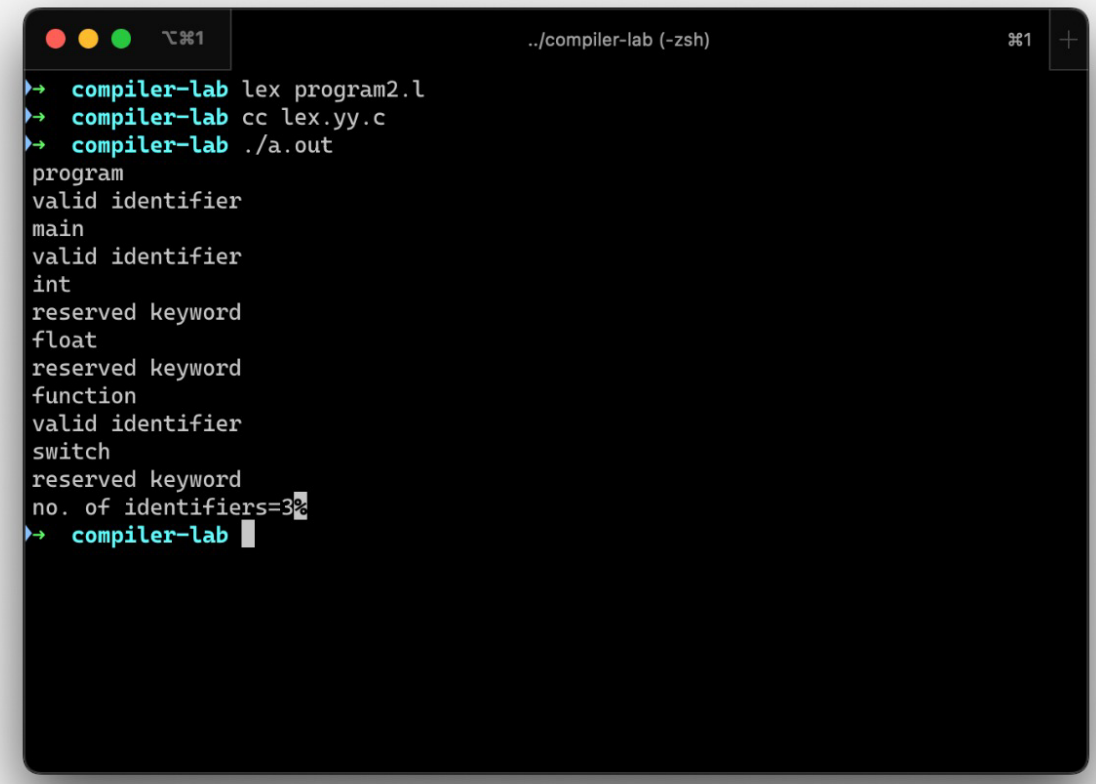
1. Write a lex program to count the number of spaces, lines, tabs and characters.

```
%{
#include<stdio.h>
int nl=0,tc=0,sc=0,ch=0;
%}
%%
\n {nl++;}
\t {tc++;}
[ ] {sc++;}
. {ch++;}
%%
int yywrap() {
    return 1;
}

int main() {
    yylex();
    printf("Number of lines: %d\n",nl);
    printf("Number of tabs: %d\n",tc);
    printf("Number of spaces: %d\n",sc);
    printf("Number of characters: %d\n",ch);
    return 0;
}
```



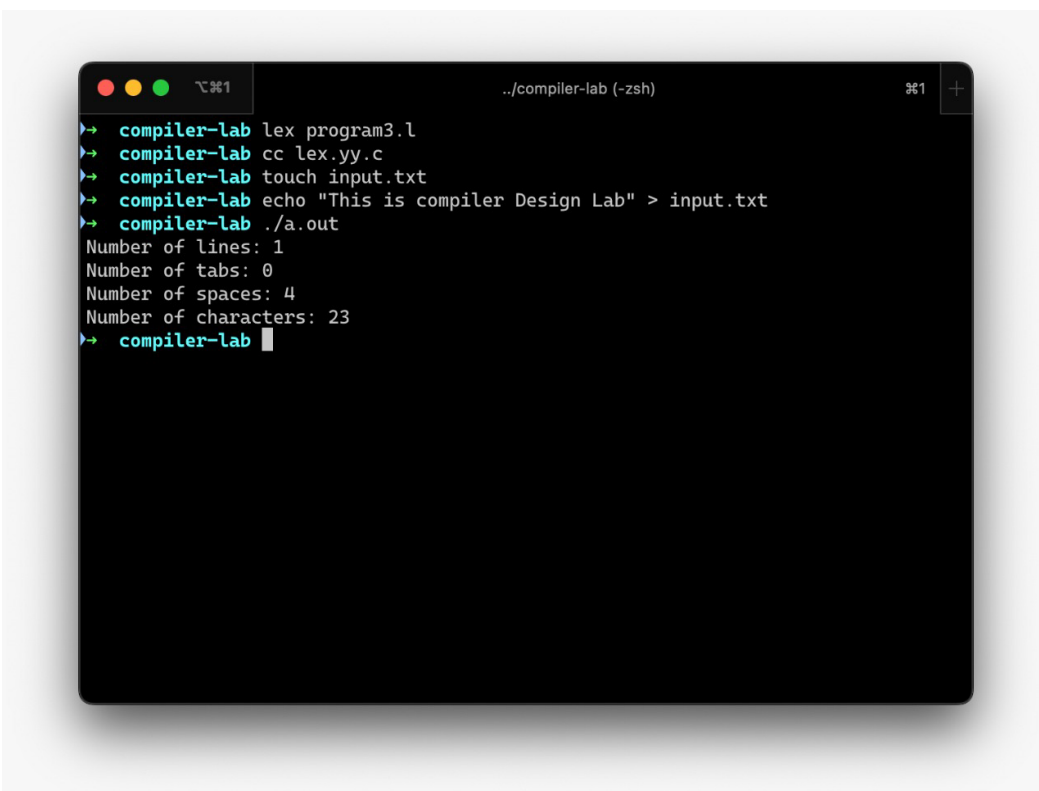
2. Write a lex program to count the number of valid identifiers.

```
%{
#include<stdio.h>
int num=0;
%}
%%
auto|else|long|switch|break|enum|register|typedef|case|extern|return|union|char|float|short|
unsigned|const|for|signed|void|continue|goto|sizeof|volatile|default|if|static|while|do|int|
struct|Packed {printf("reserved keyword");}
([a-zA-Z_][a-zA-Z0-9_]*) {num++;printf("valid identifier");}
^([0-9a-zA-Z])* {printf("not a identifier");}
%%
int yywrap() {
    return 1;
}

int main() {
    yylex();
    printf("no. of identifiers=%d",num);
}
```



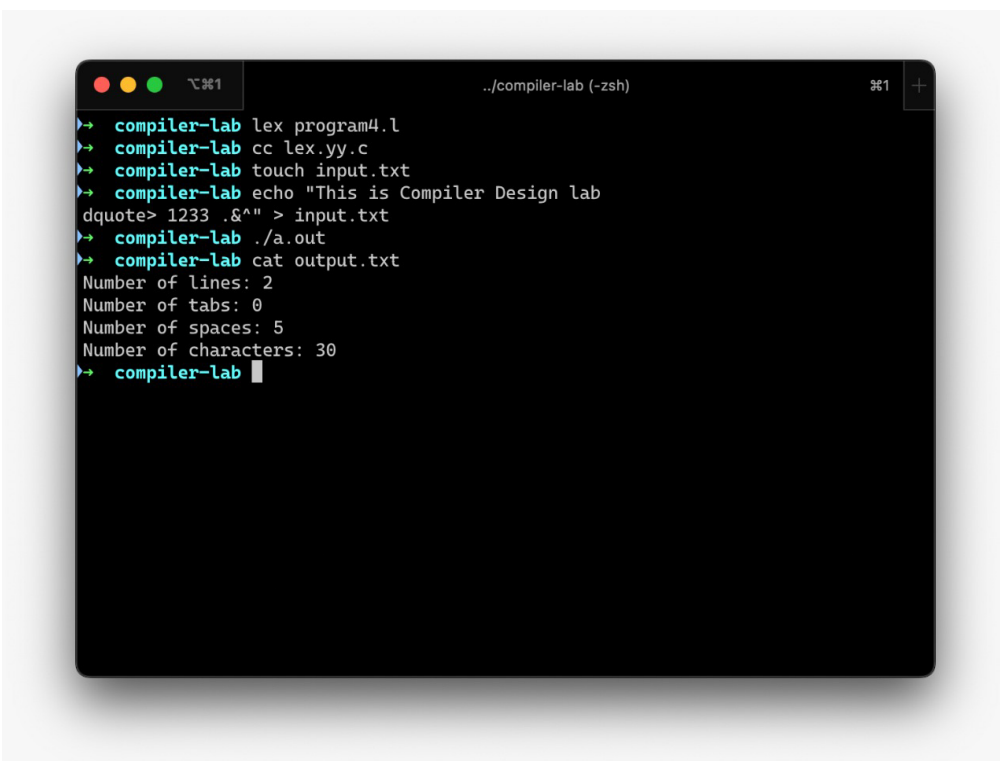
3. Write a lex program to count the number of spaces, lines, tabs and characters using file handling for input.

```
%{
#include<stdio.h>
int nl=0,tc=0,sc=0,ch=0;
%}
%%
\n {nl++;}
\t {tc++;}
[ ] {sc++;}
. {ch++;}
%%
int yywrap() {
    return 1;
}

int main() {
    extern FILE *yyin;
    yyin=fopen("input.txt","r");
    yylex();
    printf("Number of lines: %d\n",nl);
    printf("Number of tabs: %d\n",tc);
    printf("Number of spaces: %d\n",sc);
    printf("Number of characters: %d\n",ch);
    return 0;
}
```



4. Write a lex program to count the number of spaces, lines, tabs and characters using file handling for input and output.

```
%{
#include<stdio.h>
int nl=0,tc=0,sc=0,ch=0;
%}
%%
\n {nl++;}
\t {tc++;}
[ ] {sc++;}
. {ch++;}
%%
int yywrap(){
return 1;
}

int main() {
    extern FILE *yyin, *yyout;
    yyin=fopen("input.txt", "r");
    yyout=fopen("output.txt", "w");
    yylex();
    fprintf(yyout, "Number of lines: %d\n", nl);
    fprintf(yyout, "Number of tabs: %d\n", tc);
    fprintf(yyout, "Number of spaces: %d\n", sc);
    fprintf(yyout, "Number of characters: %d\n", ch);
    return 0;
}
```



5. Write a lex program to count the number of identifiers, separators, keywords, operators, and decimals using file handling.

```
%{
#include<stdio.h>
int nk=0,sp=0,op=0,id=0,integer=0,dec=0;
%}
%%
auto|else|long|switch|break|enum|register|typedef|case|extern|return|union|char|float|short|
unsigned|const|for|signed|void|continue|goto|sizeof|volatile|default|if|static|while|do|int|
struct|Packed {nk++;}
"+"|"*"|"/"|"="|"%"|"("|")"|"{"|"}" {op++;}
"."|";"|","|"-" {sp++;}
([a-zA-Z_][a-zA-Z0-9_]*) {id++;}
[^.][0-9][^.] {integer++;}
([0-9][.][0-9]) {dec++;}
%%
int yywrap() {
    return 1;
}

int main()
{
    extern FILE *yyin,*yyout;
    yyin = fopen("input.txt","r");
    yyout = fopen("output.txt","w");
    yylex();
    fprintf(yyout," No. of keyword: %d",nk);
    fprintf(yyout,"\n No. of separator: %d",sp);
    fprintf(yyout,"\n No. of operator: %d",op);
    fprintf(yyout,"\n No. of identifier: %d",id);
    fprintf(yyout,"\n No. of integer: %d",integer);
    fprintf(yyout,"\n No. of decimal: %d",dec);
    return 0;
}
```

```
compiler-lab lex program5.l
compiler-lab cc lex.yy.c
compiler-lab echo "int main ;* . 987 55.09" > input.txt
compiler-lab ./a.out
compiler-lab cat output.txt
   7 No. of keyword: 1
 No. of separator: 3
 No. of operator: 1
 No. of identifier: 1
 No. of integer: 3
 No. of decimal: 0%
compiler-lab
```

6. Write a lex program to replace the occurrence with whitespaces using file handling.

```
%{
#include<stdio.h>
int nl=0,tc=0,sc=0,ch=0;
%}
%%
([a-zA-Z]) {fprintf(yyout," ");}
%%
int yywrap() {
    return 1;
}
int main() {
    extern FILE *yyin, *yyout;
    yyin=fopen("input.txt","r");
    yyout=fopen("output.txt","w");
    yylex();
    fprintf(yyout, "Number of lines: %d\n", nl);
    fprintf(yyout, "Number of tabs: %d\n", tc);
    fprintf(yyout, "Number of spaces: %d\n", sc);
    fprintf(yyout, "Number of characters: %d\n", ch);
    return 0;
}
```