System Software Tutorial: Lex And Yacc Programs

1. LEX program to identify the given character as alphabet, number or special symbol.

```
%%
[a-z] {printf("alphabet");}
[A-Z] {printf("alphabet");}
[0-9] {printf("number");}
. {printf("special char");}
%%
 int yywrap(){
return 1;
}
 int main(){
yylex();
return 0;
}
```

```
C:\Users\balaj\ss>a.exe
a
alphabet
B
alphabet
1
number
@
special char
```

2. LEX program to identify valid decimal numbers.

```
%{
#include<stdio.h>
%}
%%
0 {printf("Valid decimal number");}
(\-)?[1-9][0-9]*(\.[0-9]*[1-9])? {printf("Valid decimal number");}
.* {printf("Invalid Decimal number");}
%%
int yywrap(){
return 1;
}
int main(){
yylex();
return 0;
}
```

```
C:\Users\balaj\ss>a.exe
1
Valid decimal number
1.1
Valid decimal number
0
Valid decimal number
-1.102
Valid decimal number
-2
Valid decimal number
```

3. LEX program to identify valid binary numbers.

```
%%
^(0|1[0-1]*((\.[0-1]*1)?)) {printf("Valid ");}
.* {printf("Invalid");}
%%
int yywrap(){
return 1;
}
int main(){
yylex();
return 0;
}
```

```
C:\Users\balaj\ss>a.exe
0110
Invalid
0
Valid
10
Valid
10010
Valid
011
Invalid
101.101
Valid
```

4. LEX program to identify valid identifiers of C programming language.

```
% {
#include <stdio.h>
% }
% %
^[a - z A - Z _][a - z A - Z 0 - 9 _] * printf("Valid Identifier");
^[^a - z A - Z _] printf("Invalid Identifier");
.;
% %
int yywrap(){
return 0;
}
 int main(){
yylex();
return 0;
}
```

```
C:\Users\balaj\ss>a.exe
1number
Invalid Identifier
_abc1
Valid Identifier
Number4
Valid Identifier
```

5. LEX program to count the number of words whose length is less than 8 but greater than 3.

```
%{
#include<stdio.h>
#include<string.h>
int len=0, count=0;
%}
%%
[a-zA-Z]+ {
        if(yyleng<8 && yyleng>3)
          {count++;}  }
%%
int yywrap (void )  {
return 1;
}
int main() {
  yylex();
  printf("%d\n", count);
  return 0; }
```

```
C:\Users\balaj\ss>a.exe
hello how are you doing today

3
^CTerminate batch job (Y/N)? y
```

6. LEX program to count only the negative numbers.

```
%{
#include<stdio.h>
  int count=0;
%}
%%
-[1-9](\.[0-9]*)? {count++;}
.
\n {printf("\n Count:%d",count);}
%%
int yywrap (void) {
    return 1;
}
int main() {
  printf("Enter the numbers:");
  yylex();
  return 0;
}
```

```
C:\Users\balaj\ss>a.exe
Enter the numbers:-1 -2.2 5 6 -7.1 1.1

 Count:3
```

7. LEX program to count number of characters, words and lines present in a file.

```
%{
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
  int l=0,w=0,c=0;
%}
%%
[a-zA-Z]*\n {l++;w++;c+=yyleng-1;}
[a-zA-Z]+(" "|\t) {w++;c+=yyleng-1;}
%%
int yywrap()
{
Return 1;
}
int main()
{
  yyin = fopen("input1.txt","r");
  yylex();
  printf("Char=%d,Words=%d,Lines=%d",c,w,l);
  return 0;
}
```

INPUT: hello there

General kenobi a pleasant surprise

```
C:\Users\balaj\ss>a.exe
Char=40,Words=7,Lines=2
C:\Users\balaj\ss>
```

8. LEX program to remove comments from a give C file and write it to another file.

```
%{
#include<stdio.h>
%}
%%
\/\/(.*) ;
\/\*(.*\n)*.*\*\/ ;
%%
int yywrap(){
return 1;
}
int main(){
yyin = fopen("inp2.c","r");
yyout = fopen("outp2.c","w");
yylex();
return 0;
}
```

INPUT:

```
#include<stdio.h>

int main(){
//Declaration /*of*/ variables
int number1,number2,result;

//Taking Inputs
printf("Enter the first number:");
scanf("%d",&number1);
printf("Enter the second number:");
scanf("%d",&number2);

/*Single line*/

/*Multiplication //of two numbers
result=number1 x number 2*/
result = number1 * number2 ;

//Displaying the data
printf("\n%d x %d = %d\n",number1,number2,result);

return 0;
}
```

OUTPUT:

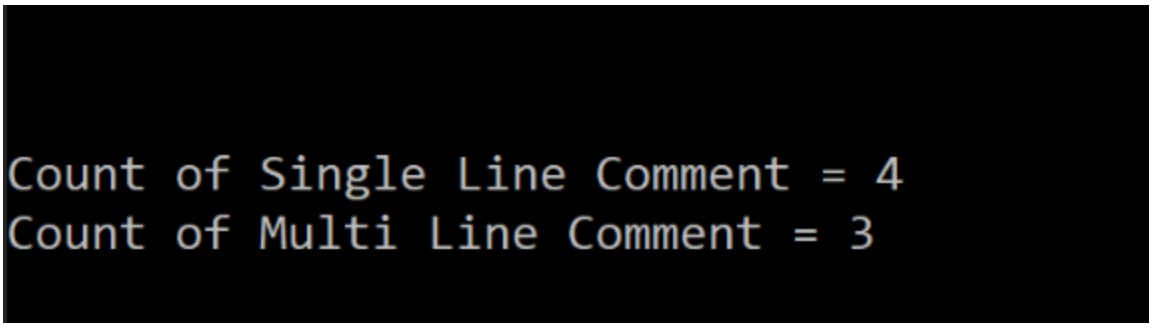
```
#include<stdio.h>

int main(){

int number1,number2,result;



printf("Enter the first number:");
scanf("%d",&number1);
printf("Enter the second number:");
scanf("%d",&number2);



result = number1 * number2 ;



printf("\n%d x %d = %d\n",number1,number2,result);

return 0;
}
```

9. LEX program to count the comments present in a file

```
%{
#include<stdio.h>
int s = 0,m = 0;
%}

%%
"//".* {s++;}
"/*"(.*)"*/" {s++;}
"/*"([^("*/")]*\n)+([^("*/")]*)"*/" {m++;}
. ;
%%

int yywrap(){
return 1;
}

int main(){
yyin = fopen("9.txt","r");
yylex();
printf("Count of Single Line Comment = %d\n",s);
printf("Count of Multi Line Comment = %d\n",m);
return 0;
}
```

INPUT:

```
#include<stdio.h>
int main(){
//This is a single line comment

/*This is a multi line comment written in a single line*/

/*This is a multi line comment
written in
multiple lines*/

/*This is 2nd multi line comment written in a single line*/

/*Second
multiline comment written
in multiple lines*/

//This is the second single line comment

/*This is a
multi line comment*/

printf("Hello World!!");
return 0;
}
```



10. LEX program to extract html tags from the given html file.

```
%%
"<"[^>]*">" {printf("%s\n",yytext);}
.
%%
int yywrap(){}
int main(){
yyin=fopen("inp3.txt","r");
yylex();
return 0;
}
```

INPUT:

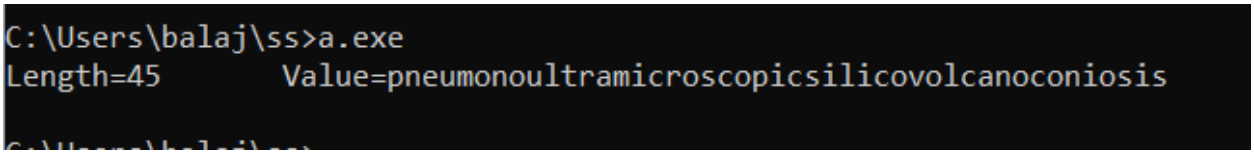
```
<HTML>
<main>
<body>
<div>hello<\div>

<p>Y?<\p>
<\body>
<\main>
```

```
C:\Users\balaj\ss>a.exe
<HTML>

<main>

<body>

<div>
<\div>


<p>
<\p>

<\body>

<\main>
```

11. LEX program to find and print the longest word in a given file with its length.

```
%{
#include<stdio.h>
#include<string.h>
int l=0;
char str[100];
%}
%%
[a-zA-Z]+ {if(yyleng>l){l=yyleng;strcpy(str,yytext);}}
. |
\n ;
%%
int yywrap(){
return 1;
}
int main(){
yyin = fopen("longword.txt","r");
yylex();
printf("Length=%d\t",l);
printf("Value=%s\n",str);
return 0;
}
```

INPUT: Add a recovery email to retain access when single-sign on is not available pneumonoultramicroscopicsilicovolcanoconiosis.



12)

LEX:

```
%{
#include "y.tab.h"
%}
%%
[0-9]+ {yylval=atoi(yytext); return INT;}
\n {return 0;}
[ \t]
. {return yytext[0];}
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token INT
%left '+''-'
%%
```

```
Expr:E{printf("Valid expr\n");return 1;}
;
E:INT'-'INT
|INT'+'INT
;
%%
int yyerror(char *msg) {
printf("Invalid expr\n");
}
int main(){
yyparse();
}
```

```
C:\Users\balaj\ss>a.exe
a+b
Invalid expr

C:\Users\balaj\ss>a.exe
1+3
Valid expr

C:\Users\balaj\ss>a.exe
4-4
Valid expr

C:\Users\balaj\ss>a.exe
1-
Invalid expr
```

13)

LEX:

```
%{
#include "y.tab.h"
%}
%%
[0-9]+ {yylval=atoi(yytext); return INT;}
\n {return 0;}
[ \t]
. {return yytext[0];}
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token INT
%left '+''-'
%left '*''/'
%%
Expr:E{printf("Valid expr\nValue=%d\n",$1);return 1;}
;
```

```
E:E'*'E {$$=$1*$3;}
|E'/'E {$$=$1/$3;}
|E'-'E {$$=$1-$3;}
|E'+'E {$$=$1+$3;}
|INT
;
%%
int yyerror(char *msg) {
printf("Invalid expr\n");
}
int main(){
yyparse();
}
```

```
C:\Users\balaj\ss>a.exe
3+4*2
Valid expr
Value=11

C:\Users\balaj\ss>a.exe
8-3/3-1
Valid expr
Value=6
```

14)

LEX:

```
%{
#include"y.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token A B NL
%start S
%%
S:A A A X B B B NL {printf("Valid\n");return 1;}
;
X:A X B
|
```

```
;
%%
int yyerror(char *msg) {
printf("Invalid string\n");
}
int main(){
yyparse();
}
```

```
C:\Users\balaj\ss>a.exe
aaabbb
Valid

C:\Users\balaj\ss>a.exe
aab
Invalid string

C:\Users\balaj\ss>a.exe
aaaabbb
Invalid string

C:\Users\balaj\ss>a.exe
aaaabbbb
Valid
```

15)

LEX:

```
%{
#include"y.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token A B NL
%start S
%%
S:A X A NL {printf("Valid\n");return 1;}
;
X:B X
|
;
```

```
%%
int yyerror(char *msg) {
printf("Invalid string\n");
}
int main(){
yyparse();
}
```

```
C:\Users\balaj\ss>a.exe
aa
Valid

C:\Users\balaj\ss>a.exe
abbaa
Invalid string

C:\Users\balaj\ss>a.exe
abbbba
Valid
```

16)

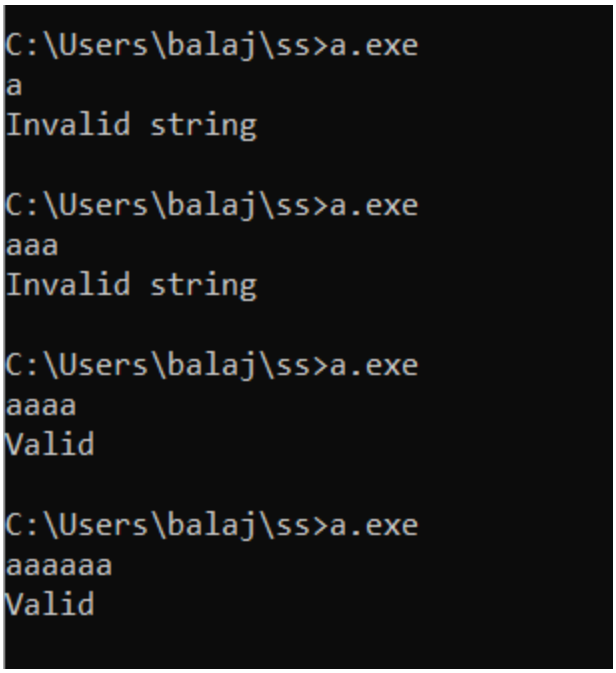
LEX:

```
%{
#include"y.tab.h"
%}
%%
[aA] {return A;}
\n {return NL;}
. {return yytext[0];}
```

```
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token A NL
%start S
%%
S:X NL {printf("Valid\n");return 1;}
;
X:X A A
|A A
;
%%
int yyerror(char *msg) {
printf("Invalid string\n");
}
int main(){
yyparse();
}
```



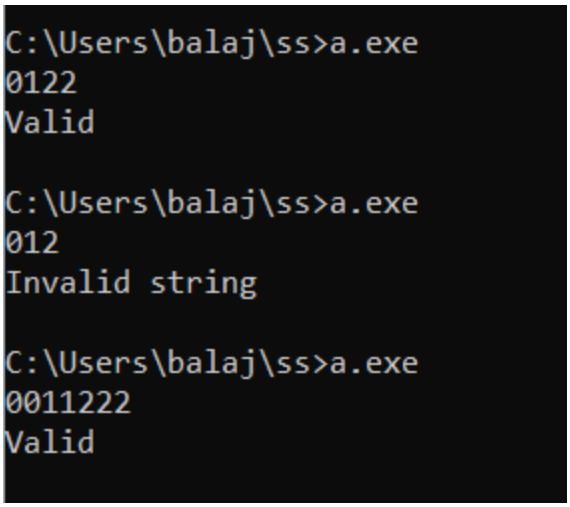
17)

LEX:

```
%{
#include"y.tab.h"
%}
%%
[0] {return A;}
[1] {return B;}
[2] {return C;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token A B NL C
%start S
%%
S:X Y NL {printf("Valid\n");return 1;}
;
X: A X B
|A B
;
Y:Y C
|C C
;
%%
int yyerror(char *msg) {
printf("Invalid string\n");
}
int main(){
yyparse();
}
```



18)

LEX:

```
%{
#include"y.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
[cC] {return C;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap(){}
```

YACC:

```
%{
#include<stdio.h>
%}
%token A B NL C
%start S
%%
S:B X B {printf("Valid");return 1;}
;
X: B X B|A X A|C
%%
int yyerror(char *msg) {
printf("Invalid string\n");
}
int main(){
yyparse();
}
```

```
C:\Users\balaj\ss>a.exe
bcb
Valid
C:\Users\balaj\ss>a.exe
abcba
Invalid string

C:\Users\balaj\ss>a.exe
bacab
Valid
C:\Users\balaj\ss>a.exe
bbbaacaabbb
Valid
```