Lex Lab Programs

Input to Lex is divided into three sections with %% dividing the sections

... definitions ...

%%

... rules ...

%%

... subroutines ...

Name	Function
int yylex(void)	call to invoke lexer, returns token
char *yytext	pointer to matched string
yyleng	length of matched string
yylval	value associated with token
int yywrap(void)	wrapup, return 1 if done, 0 if not done
FILE *yyout	output file
FILE *yyin	input file
ECHO	write matched string

```
%{
#include<stdio.h>
int v=0,c=0;
%}
%%
[ \t\n]+ ;
[aeiouAEIOU] {v++;}
[^aeiouAEIOU] {c++;}
```

```
%%
int main()
{
printf("\n Enter the input\n");
yylex();
printf("number of vowels is %d\n",v);
 printf("number of consonants is %d\n",c);
}

int yywrap()
{ return 1;
}
```

```
%%              /* match everything except newline */
. ECHO;         /* match newline */
 \n ECHO;
 %%

int yywrap(void)
 {
return 1;
}
int main(void)
 {
yylex(); /*main entry point for lex*/
 return 0;
```

To see the output

cmd

flex 1.l

gcc lex.yy.c

a.exe

1. **/* lex program to count the number of vowels and consonants in a given string*/**

```
%{
#include<stdio.h>
int v=0,c=0;
%}
%%
[ \t\n]+;
[aeiouAEIOU] {v++;}
[^aeiouAEIOU] {c++;}
%%
int main()
{
printf("\n Enter the input\n");
yylex();
printf("number of vowels is %d\n",v);
 printf("number of consonants is %d\n",c);
}

 int yywrap()
 { return 1;
 }
```

Space

2. **/* lex program to count the number of words, characters, lines and spaces in a given input*/**

```
%{
#include<stdio.h>
int lc=0,sc=0,wc=0, cc=0;
%}
%%
[\n] {lc++; cc+=yyleng;}
[ \t] { sc++; cc+=yyleng;}
[^\t\n ]+ {wc++; cc+=yyleng;}
%%
int main()
{
printf("\n Enter the input\n");
yylex();
printf("number of words is %d\n", wc);
printf("number of characters is %d\n",cc);
 printf("number of lines is %d\n", lc);
printf("number of space is %d\n", sc);
}

 int yywrap()
 { return 1;
 }
```

3. **/* lex program to count the number of characters and digits in a given input*/**

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

```
%%
[^\t\n ]+;  /* ignore these i.e new, line, space, tab*/
[a-zA-Z] {c++;}
[0-9] {n++;}
%%
int main()
{
printf("\n Enter the input\n");
yylex();
printf("number of characters %d\n",c);
 printf("number of digits is %d\n",n);
}

int yywrap()
{ return 1;
}
```

Output

```
Enter the input
anupama 10

^Z
number of characters 7
number of digits is 2
```

4. **/* lex program to identify and count positive and negative numbers*/**

```
%{
#include<stdio.h>
int p=0,n=0;
```

```
%}
%%
[0-9]+ {p++;}
[-][0-9]+ {n++;}
%%
int main()
{
printf("\n Enter the input\n");
yylex();
printf("number of positive number %d\n",p);
printf("number of negative number is %d\n",n);
}
int yywrap()
{ return 1;
}
```

```
Enter the input
3 -9 4 -3 -5 -6 2

^Z
number of positive number 3
number of negative number is 4
```

5. /* lex program to identify capital words in a given input*/

```
%{
#include<stdio.h>
%}
%%
[A-Z]+[ \t\n] {printf("%s\n", yytext);}
. ;
```

```
%%
int main()
{
printf("\n Enter the string\n");
yylex();
}

int yywrap()
{ return 1;
}
```

output

```
Enter the string
my name is ANUPAMA PRASHANTH
ANUPAMA
PRASHANTH
```

6. **/* lex program to identify keywords, numbers and words in a given input*/**

```
%{
#include<stdio.h>
%}
%%
if |
else |
printf {printf("%s is a keyword", yytext);}
[0-9]+ {printf("%s is a number", yytext);}
[a-zA-Z]+ {printf("%s is a word", yytext);}
.|\n {ECHO;}
%%
int main()
```

```
{
printf("\n Enter the string\n");
yylex();
}
    int yywrap()
    { return 1;
    }
```

Output

Enter the string

anupama 10 if

anupama is a word 10 is a number if is a keyword