LEX SIMPLE PROGRAMS

SIMPLE ADDITION PROGRAM USING LEXTOOL

PROGRAM:

```
%{
#include<stdio.h>
int a,b,c;
%}
%%
“a”  printf(“enter the value of a:”);  scanf(“%d”,&a);
“b”  printf(“enter the value of b:”);  scanf(“%d”,&b);
“c”  printf(“the addition of %d,%d is %d:”,a,b,c=a+b);
%%
int main()
{
  yylex();
  return 0;
}
int yywrap()
{
  return 0;
}
```

INPUT:

```
 vi filename.l
 lex filename.l
 cc lex.yy.c
 ./a.out
```

OUTPUT:

```
a
enter the value of a:45

b
enter the value of b:35

c
the addition of 45,35 is 80
```

COUNTING VOWELS ,LETTERS AND DIGITS

PROGRAM:

```
%{
#include<stdio.h>
int vow=0,num=0,let=0;
%}
```

```
%%
[aeiouAEIOU] vow++;
[0-9] num++;
[A-Za-z] let++;
";" printf("\nVOWELS=%d,LETTERS=%d,DIGITS=%d",vow,let,num);
%%
int main()
{
yylex();
return 0;
}
int yywrap()
{
return 0;
}
```

INPUT:

```
vi filename.l
lex filename.l
cc lex.yy.c
./a.out
```

OUTPUT:

```
hello1
;
VOWELS=2,LETTERS=3,DIGITS=1
```

SIMPLE PROGRAM USING LEX TOOL

PROGRAM:

```
%{

%}

%%

"rama" |

"seetha" |

"geetha" printf("\n noun");

"sings" |
```

```
"dances" |

"eats" printf("\n verb");

"perfectly" |

"nicely" |

"loudly" printf("\n adjective");

%%

int main()

{

 yylex();

 return 0;

}

int yywrap()

{

 return 1;

}
```

OUTPUT:

```
lex x.l

cc lex.yy.c

./a.out


                seetha

                     noun

                dances
```

verb

perfectly

adjective

SIMPLE PROGRAM TO FIND THE NUMBER OF CHARACTERS,WORDS,LINES USING LEX TOOL

PROGRAM:

```
%{
   int char_cnt=0,word_cnt=0,line_cnt=0;
%}
word [^ \t \n]+
%%
{word}     {word_cnt++;char_cnt+=yyleng;} \n   {char_cnt++;line_cnt++;}
. char_cnt++;
%%
int main(int argc,char **argv)
{
  if(argc  > 1)
  {
    FILE *file;
    file=fopen(argv[1],"r");
    if(!file)
     {
```

```
        fprintf(stderr,"could not open %s\n",argv[1]);
        exit(1);
    }
  yyin= file;/*standard input file*/
}
yylex();
printf("%d %d %d \n",char_cnt,word_cnt,line_cnt);
return 0;
}
int yywrap()
{
return 1;
}
```

INPUT:

```
vi aa.c
compiler program is
 very elaborative.
```

OUTPUT:

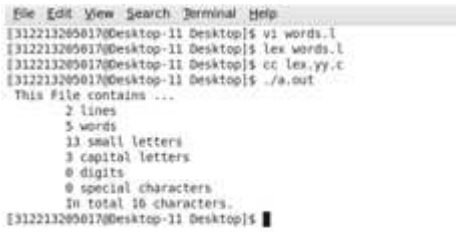
```
lex number.l
cc lex.yy.c
./a.out aa.c
39      5       2
```

A PROGRAM TO FIND THE NUMBER OF LINES, WORDS,LETTERS AND SPECIAL CHARACTERS USING LEX TOOL

Program

```
%{include<stdio.h>
int lines=0, words=0,s_letters=0,c_letters=0, num=0, spl_char=0,total=0;
%}
%%



\n { lines++; words++;}
[\t ' '] words++;
[A-Z] c_letters++;
[a-z] s_letters++;
[0-9] num++;
. spl_char++;
%%




main(void)
{
yyin= fopen("myfile.txt","r");
yylex();
total=s_letters+c_letters+num+spl_char;
printf(" This File contains ...");
printf("\n\t%d lines", lines);
printf("\n\t%dwords",words);
printf("\n\t%d small letters", s_letters);
printf("\n\t%d capital letters",c_letters);
printf("\n\t%d digits", num);
printf("\n\t%d special characters",spl_char);
printf("\n\tIn total %d characters.\n",total);
}

intyywrap()
{
return(1);}
```

Output



LEXICAL ANALYSER USING LEXTOOL

PROGRAM:

```
%{
 #include<stdio.h>
 #include<string.h>
 char str[20];
 int opc=0,dc=0,lc=0,varc=0,idc=0;
%}
digit [0-9]
op [%|*|+|-|=]
id [for|if|while|do|const|break|default]

%%
 {digit} dc++;
 {op} opc++;
 {id} idc++;
“\n” printf(“digit %d\noperator%d,\nidentifier%d,dc,opc,idc);
%%
int main()
{
yylex()
return 0;
}
int yywrap()
```

```
{
return 0;
}
```

INPUT:

```
lex filename.l
cc lex.yy.c
./a.out
```

OUTPUT:

```
2+r=p
digit 1
operator 2,
identifier 1
```