Logo, company name

Description automatically generated

**COMPILER DESIGN LAB**

**PCS – 601**

**MID TERM PRACTICAL**

**Submitted By:**

Sahil Obhrai

University Roll No.: 2015270

B.Tech CSE (DS & AI)

Section: DS

Class Roll No.: 14

**Submitted To:**

Mrs. Vishu Tyagi

**Q1. Design a LEX Code to count the number of lines, space, tab-meta character, and rest of characters in each Input pattern.**

**ANSWER.**

%{  
#include<stdio.h>  
int tab=0, space=0,line=0, ch=0;  
%}  
  
%%  
"\n" line++;  
" " space++;  
"\t" tab++;  
([a-zA-Z0-9])+ ch++;  
  
%%  
  
int yywrap(void){}  
  
void main()  
{  
yylex();  
  
printf("\nNo of lines %d",line);  
printf("\nNo of spaces: %d",space);  
printf("\nNo of tabs: %d",tab);  
printf("\nNo of characters: %d",ch);  
printf("\n");  
}

**OUTPUT:**

**A picture containing chart

Description automatically generated**

**Q2. Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.**

**ANSWER:**

%{  
#include<stdio.h>  
#include<string.h>  
%}  
  
%%  
^[a-zA-Z\_][a-zA-Z0-9]\* printf("Valid Identifier ");  
.\* printf("Invalid Identifier");  
%%

int yywrap(){return 0;}  
void main()  
{  
yylex();  
}

**OUTPUT:**

**A picture containing shape

Description automatically generated**

**Q3. Design a LEX Code to identify and print integer and float value in given Input pattern.**

**ANSWER:**

%{  
#include<stdio.h>  
%}

%%  
[0-9]+ printf("Integer Value\n");  
[0-9]+"."[0-9]+ printf("Float value\n");  
.\* {printf("Invalid Value\n");}  
%%  
  
int yywrap() {return 0;}  
  
void main()  
{  
yylex();  
}

**OUTPUT:**

**Shape

Description automatically generated with low confidence**

**Q4. Design a LEX Code to count and print the number of total characters, words, white spaces in given ‘Input.txt’ file.**

int p=1,d=0,r=4;

float m=0.0, n=200.0.

while (p <= 3)

{ if(d==0)

{ m= m+n\*r+4.5; d++; }

else

{ r++; m=m+r+1000.0; }

p++; }

**ANSWER:**

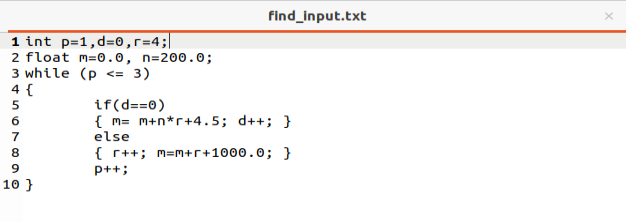
%{  
#include<stdio.h>  
%}  
  
%%  
"int"|"float"|"double"|"while"|"if"|"else"|"return"|"main" { fprintf(yyout,"\nKeyword: %s",yytext); }  
^[a-zA-Z\_][a-zA-Z0-9]\* { fprintf(yyout,"\nIdentifier %s",yytext); }  
"+"|"++"|"-"|"--"|"="|"=="|">="|"<="|"\*"|"/" { fprintf(yyout,"\nOperator: %s",yytext); }  
[{}(), ;] { fprintf(yyout,"\nSeparator: %s",yytext); }  
\n { fprintf(yyout,"\n"); }  
. ;  
%%

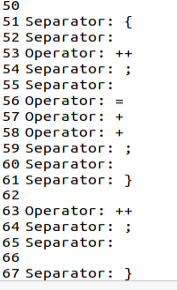
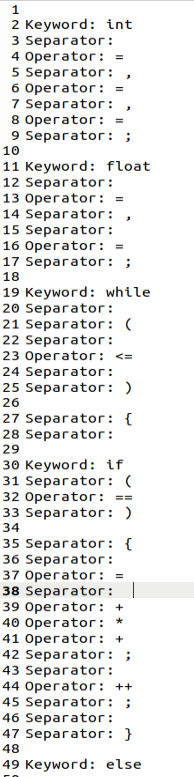
int yywrap(){return 1;}

int main(void)  
{  
extern FILE \*yyin, \*yyout;  
yyin=fopen("find\_input.txt","r");  
yyout=fopen("find\_output.txt","w");  
yylex();  
}

**OUTPUT:**

**Input File:**

****

**Output File:**

**Q5. Design a LEX Code to count and print the number of total characters, words, white spaces in given ‘Input.txt’ file.**

**ANSWER:**

%{  
#include<stdio.h>  
int tab=0, space=0,line=0,ch=0;  
%}  
  
%%  
\n line++;  
([a-zA-Z0-9])+ ch++;  
([ ])+ space++;  
\t tab++;  
%%  
  
int yywrap(void){}  
  
void main()  
{  
extern FILE \*yyin, \*yyout;  
yyin=fopen("input.txt","r");  
yyout=fopen("output.txt","w");  
  
yylex();  
  
fprintf(yyout,"\nNo of Characters: %d",ch);  
fprintf(yyout,"\nNo of lines %d",line+1);  
fprintf(yyout,"\nNo of spaces: %d",space);  
fprintf(yyout,"\nNo of tabs: %d",tab);   
}

**OUTPUT:**

**Input File:**

**A picture containing table

Description automatically generated**

**Output File:**

Table

Description automatically generated

**Q6. Design a LEX Code to replace white spaces of ‘Input.txt’ file by a single blank character into ‘Output.txt’ file.**

**ANSWER:**

%{  
#include <stdio.h>  
%}  
  
%%  
[ \n\t]+ {fprintf(yyout," ");}  
. {fprintf(yyout,"%s",yytext);}  
%%  
  
int yywrap(void){}  
  
int main()  
{  
extern FILE \*yyin, \*yyout;  
yyin=fopen("Inputrw.txt","r");  
yyout=fopen("Output.txt","w");  
  
yylex();  
  
}

**OUTPUT:**

**Input File:**

**Graphical user interface, application, Word

Description automatically generated**

**Output File:**

**Graphical user interface, text

Description automatically generated with medium confidence**

**Q7. Design a LEX Code to remove the comments from any C-Program given at run-time and store into ‘out.c’ file.**

**ANSWER:**

%{  
#include <stdio.h>  
%}  
  
%%  
\/\/(.\*) {};  
\/\\*(.\*\n)\*.\*\\*\/ {};  
  
%%  
  
int yywrap(void) {}  
  
int main()  
{  
extern FILE \*yyin, \*yyout;  
yyin=fopen("in.c","r");  
yyout=fopen("out.c","w");  
  
yylex();  
  
}

**OUTPUT:**

**Input File:**

**Graphical user interface, text, application

Description automatically generated**

**Output File:**

**Text

Description automatically generated**

**Q8. Design a LEX Code to extract all html tags in the given HTML file at run time and store into Text file given at run time.**

**ANSWER:**

%{  
#include<stdio.h>  
%}  
  
%%  
"<"[^>]\*> {fprintf(yyout,"%s\n",yytext);}  
. ;  
%%  
  
int yywrap(void){}  
  
void main()  
{  
extern FILE \*yyin, \*yyout;  
yyin=fopen("tags.html","r");  
yyout=fopen("onlytags.txt","w");  
  
yylex();  
  
}

**OUTPUT:**

**Input File:**

**Text

Description automatically generated**

**Output File:**

**Graphical user interface, application

Description automatically generated**