**Name: Ranjana Singh Roll No:46 Section:E**

**Program 1:** Design a LEX Code to count the number of lines, space, tab-meta character

and rest of Characters in a given input patterns.

**Solution:**

%{

#include<stdio.h>

int l=0,sp=0,tab=0,c=0;

%}

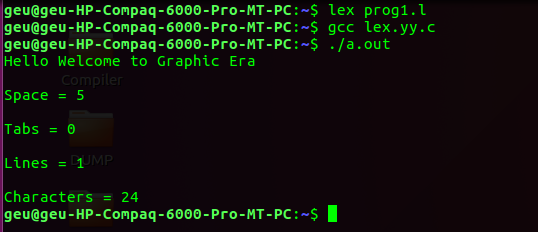
\n l++;

[ ] sp++;

\t tab++;

. c++;

| %% | |
| --- | --- |
| int yywrap()  int main(){  yylex();  printf("\nSpace=%d\n”,sp);  printf("\nTabs = %d\n",tab);  printf("\nLines = %d\n",l);   | printf("\nCharacters = %d\n",c); | | --- |   }  **OUTPUT:** | |



|  |
| --- |

|  |
| --- |

**Name: Ranjana Singh Roll No:46 Section:E**

**Program 2:**Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.

**Solution:**

%{

int count=0;

%}

op [+-\*/]

letter [a-zA-Z]

digitt [0-9]

id {letter}\*|({letter}{digitt})+

notid ({digitt}{letter})+

%%

[\t\n]+

("int")|("float")|("char")|("case")|("default")|("if")|("for")|("printf")|("scanf") {printf("%s is a keyword\n", yytext);}

{id} {printf("%s is an identifier\n", yytext); count++;}

{notid} {printf("%s is not an identifier\n", yytext);}

%%

yywrap(){

}

int main(int argc, char \*argv[]){

extern FILE \*yyin;

yyin = fopen(argv[1],"r");

yylex();

printf("TOTAL IDENTIFIERS %d",count);

return 0;

}

**OUTPUT:**



**Name: Ranjana Singh Roll No:46 Section:E**

**Program 3:**Design a LEX Code to identify and print integer and float value in given Input pattern.

**Solution:**

%{

%}

DIGIT [0-9]

%%

{DIGIT}\* {ECHO;printf(" Integer");}

{DIGIT}\*?\.{DIGIT}\* {ECHO;printf(" Float ");}

%%

yywrap(){

}

main(int argc, char \*argv[])

{

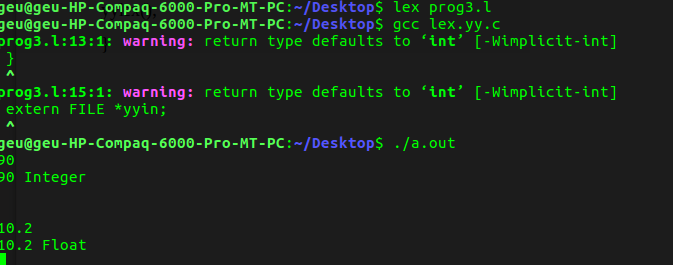
extern FILE \*yyin;

yyin = fopen(argv[1],"r");

yylex();

}

**OUTPUT:**



**Name: Ranjana Singh Roll No:46 Section:E**

**Program 4:**Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPERATORS,

KEYWORDS,IDENTIFIERS) the following C-fragments.

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++; }

**Solution:**

%{

#include <stdio.h>

%}

DIGIT [0-9] NUMBER {DIGIT}+

REAL {DIGIT}\*"."{DIGIT}+ TEXT [a-zA-Z ]+ TEXT\_NUMBERS [a-zA-Z0-9]

CONDITIONALS "if"|"else"|"else if"|"switch"|"case"

KEYWORD

"break"|"continue"|"goto"|"printf"|"scanf"|"sprintf"|"sscanf"|"fopen"|"fwrite"|"fread"|"fclose"|" write"|"read"|"open"|"close"|"return"|"int"|"float"|"char"|"unsigned"|"signed"|"short"|"long"|"d ouble"

ITERATORS "for"|"while"|"do" PREPROCESSOR

"#"|"#line"|"#undef"|"#error"|"#elif"|"#else"|"#endif"|"#if"|"#define"|"#include"|"#pragma"|"#i

fndef"|"#ifdef" DELIMITER [; :\t\n()"]

IDENTIFIER [a-zA-Z]{TEXT\_NUMBERS}\*|[a-zA- Z]{TEXT\_NUMBERS}\*[[{NUMBER}+]]

FORMAT\_SPECIFIER "%"{TEXT\_NUMBERS}+ FILE "<"{IDENTIFIER}.h">"

COMMENT "/\*"[a-zA-Z0-9 \t\n;.~!@#$%^&\*()\_+=<>?:"{}]\*"\*/" AOPERATOR "+"|"-"|"\*"|"/"|"="

BLOCK\_BEGINS "{" BLOCK\_ENDS "}" UNARY "++"|"--"

LOPERATOR "&"|"|"|"&&"|"~"|"||"|">"|"<"|">="|"<="|"=="

FUNCTION

{IDENTIFIER}+"("{DELIMITER}\*{TEXT}{TEXT\_NUMBERS}\*{DELIMITER}\*")"

%%

{CONDITIONALS} { printf("%s is a conditional\n", yytext); }

{ITERATORS} { printf("%s is an iterator\n", yytext); }

{DIGIT} { printf("%s is a digit\n", yytext); }

{NUMBER} { printf("%s is a number\n", yytext); }

{REAL} { printf("%s is a real number\n", yytext); }

{PREPROCESSOR} { printf("%s is a preprocessor directive\n", yytext); }

{DELIMITER} { printf("%s is a delimiter\n", yytext); }

{KEYWORD} { printf("%s is a keyword\n", yytext); }

{IDENTIFIER} { printf("%s is an identifier\n", yytext); }

{COMMENT} { printf("%s is a comment\n", yytext); }

{AOPERATOR} { printf("%s is a mathematical operator\n", yytext); }

{LOPERATOR} { printf("%s is a logical operator\n", yytext); }

{BLOCK\_BEGINS} { printf("Block begins\n", yytext); }

{BLOCK\_ENDS} { printf("Block ends\n", yytext); }

{FILE} { printf("%s is a file\n", yytext); }

{UNARY} { printf("%s is a unary operator\n", yytext); }

{FUNCTION} { printf("%s is a function\n", yytext); }

{FORMAT\_SPECIFIER} {printf("%s is a format specifier\n", yytext); }

%%

yywrap(){}

int main(int argc, char \*argv[]) { extern FILE \*yyin;

yyin = fopen(argv[1],"r"); yylex();

return 0;

}

**Name: Ranjana Singh Roll No:46 Section:E**

**Program 5:**Design a LEX Code to count and print the number of total characters, words, white

Spaces in given ‘Input.txt’ file.

**Solution:**

%{

#include<stdio.h>

#include<string.h>

int tchar=0,tword=0,tspace=0,ln;

%}

%%

" " {tspace++;}

[\n] {ln++;}

[a-zA-Z]\* {tword++; tchar+=strlen(yytext);}

%%

yywrap(){

}

int main(){

yyin = fopen("Input.txt","r");

yylex();

printf("Characters: %d\n Words:%d\n Spaces: %d\nLines: %d\n",tchar,tword,tspace,ln);

return 0;

}

**OUTPUT:**

