.

**🔹 1. Token Identifier for a Simple C Program**

**What it detects:**

* Keywords
* Identifiers
* Numbers
* Operators
* Punctuations

✅ *Important for lab exams*

lex

CopyEdit

%{

#include <stdio.h>

%}

IDENTIFIER [a-zA-Z\_][a-zA-Z0-9\_]\*

KEYWORDS "int"|"float"|"if"|"else"|"return"

NUMBER [0-9]+

OPERATOR [+\-\*/=<>!]=?|&&|\|\|

PUNCTUATION [\{\}\(\)\;\,\[\]]

%%

{KEYWORDS} { printf("Keyword: %s\n", yytext); }

{IDENTIFIER} { printf("Identifier: %s\n", yytext); }

{NUMBER} { printf("Number: %s\n", yytext); }

{OPERATOR} { printf("Operator: %s\n", yytext); }

{PUNCTUATION} { printf("Punctuation: %s\n", yytext); }

[ \t\n] { /\* skip whitespace \*/ }

. { printf("Unknown: %s\n", yytext); }

%%

int yywrap() { return 1; }

**🔹 2. Valid Email Detection**

Matches email formats like name@domain.com.

lex

CopyEdit

%%

[a-zA-Z0-9\_.]+@[a-zA-Z]+\.[a-zA-Z]{2,} { printf("Valid Email: %s\n", yytext); }

.|\n ;

%%

**🔹 3. Valid Mobile Number Checker (India Format)**

Detects 10-digit numbers starting with 7, 8, or 9.

lex

CopyEdit

%%

[7-9][0-9]{9} { printf("Valid Indian Mobile Number: %s\n", yytext); }

.|\n ;

%%

**🔹 4. Count Vowels, Consonants, Digits, and Special Characters**

lex

CopyEdit

%{

int v = 0, c = 0, d = 0, s = 0;

%}

%%

[aAeEiIoOuU] { v++; }

[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z] { c++; }

[0-9] { d++; }

[^a-zA-Z0-9\n] { s++; }

\n {

printf("Vowels: %d, Consonants: %d, Digits: %d, Specials: %d\n", v, c, d, s);

}

. ;

%%

**🔹 5. Find and Count All Identifiers in a Program**

lex

CopyEdit

%%

[a-zA-Z\_][a-zA-Z0-9\_]\* { printf("Identifier: %s\n", yytext); }

[ \t\n] ;

. ;

%%

**🔹 6. Detect Comments in a C Program**

lex

CopyEdit

%%

"//".\* { printf("Single-line comment: %s\n", yytext); }

"/\*"([^\*]|\\*+[^\*/])\*\\*+"/" { printf("Multi-line comment: %s\n", yytext); }

.|\n ;

%%

**🔹 7. Detect Integers, Real Numbers, and Scientific Notation**

lex

CopyEdit

%%

[0-9]+ { printf("Integer: %s\n", yytext); }

[0-9]\*\.[0-9]+ { printf("Real Number: %s\n", yytext); }

[0-9]+[eE][+-]?[0-9]+ { printf("Scientific Notation: %s\n", yytext); }

.|\n ;

%%

**🔹 8. Lex Program to Detect Valid Identifiers (as per C rules)**

lex

CopyEdit

%%

[a-zA-Z\_][a-zA-Z0-9\_]\* { printf("Valid Identifier: %s\n", yytext); }

[^a-zA-Z0-9\_\n]+ { printf("Invalid Identifier Part: %s\n", yytext); }

.|\n ;

%%

**🔹 9. Detect Function Definitions (with 0, 1, or multiple arguments)**

lex

CopyEdit

%%

[a-zA-Z\_][a-zA-Z0-9\_]\*[ \t]\*\([ \t]\*\) { printf("Function with 0 args: %s\n", yytext); }

[a-zA-Z\_][a-zA-Z0-9\_]\*[ \t]\*\([ \t]\*[a-zA-Z0-9\_]+[ \t]\*\) { printf("Function with 1 arg: %s\n", yytext); }

[a-zA-Z\_][a-zA-Z0-9\_]\*[ \t]\*\([^\)]\*,[^\)]\*\) { printf("Function with n args: %s\n", yytext); }

.|\n ;

%%

**🧠 Pro Tips for Lex Lab Evaluation**

* Always include:

c

CopyEdit

int yywrap() { return 1; }

* Use gcc lex.yy.c -lfl or gcc lex.yy.c -lfl -lm if you use math.h
* Know basic regex:
  + [a-z] → lowercase letters
  + [0-9]+ → number
  + \. → literal dot
  + \*, +, ?, {} → repetitions