**PROGRAM – 6**

**AIM :** WAP to identify keywords , operators , identifiers , delimiters , literals etc. using LEX tool .

**LEX PROGRAM :**

keyword alignas|alignof|and|and\_eq|asm|atomic\_cancel|…

operator (\:\:)|(\+\+)|(\-\-)|(\.)|(\-\>)|(\~)|(\!)|(\\*)|(\.\\*)|(\-\>\\*)|(\\*)|(\/)|…

reserve std|main|cin|cout

delimiter [,;]

bracket [(){}]

%{

#include <iostream>

using namespace std ;

int line\_no = 1 ;

%}

%%

\n {

line\_no ++ ;

cout << "\n" ;

}

\"[^\"]\*\" {

cout << yytext << "\tLiteral at Line:" << line\_no << "\n" ;

}

#.\* {

cout << yytext << " : Macro at Line:" << line\_no << "\n" ;

}

{keyword} {

cout << yytext << " : Keyword at Line:" << line\_no << "\n" ;

}

{reserve} {

cout << yytext << " : Reserved word at Line:" << line\_no << "\n" ;

}

{operator} {

cout << yytext << " : Operator at Line:" << line\_no << "\n" ;

}

{delimiter} {

cout << yytext << " : Delimiter at Line:" << line\_no << "\n" ;

}

{bracket} {

cout << yytext << " : Bracket at Line:" << line\_no << "\n" ;

}

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

cout << yytext << " : Valid identifier at Line:" << line\_no << "\n" ;

}

[ \t]\* ;

[0-9]\*[a-zA-Z\_][a-zA-Z0-9\_]\* {

cout << yytext << " : Invalid identifier at Line:" << line\_no << "\n" ;

}

[0-9]\* {

cout << yytext << " : Constant at Line:" << line\_no << "\n" ;

}

. {

cout << yytext << " : Invalid identifier at Line:" << line\_no << "\n" ;

}

%%

int yywrap() {

return 1 ;

}

int main() {

yyin = fopen("test4.cpp" , "r") ;

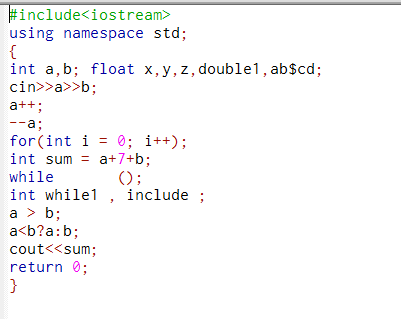
yylex() ;

fclose(yyin) ;

return 0 ;

}

**Input File :**



**Output File :**

