**Practical – 5**

**Aim:**

**(a) Write a FLEX program to print multiline comment from C file.**

**(b) Write a FLEX program to take input from text file and count no of characters, no. of spaces, no. of lines & no. of words.**

**(c) Write a FLEX program to classify operators, keywords, identifiers and separators and store it in different files.**

**Code (a):**

%{

#include <stdio.h>

%}

%s BRACKET

%%

"/\*" BEGIN(BRACKET);

<BRACKET>.|\n ECHO;

<BRACKET>"\*/" BEGIN(INITIAL);

.|\n ;

%%

void main()

{

yyin = fopen("test.c", "r");

yylex();

fclose(yyin);

}

**test.c**

#include <stdio.h>

/\*\* main program start

\* this is a dummy program

\* which does absolutely nothing

\*/

Write a FLEX program to print multiline comment from C file.

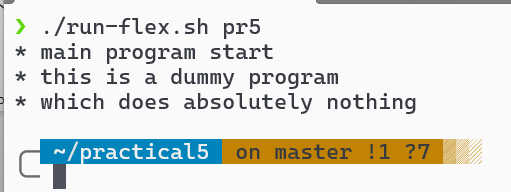
int main()

{

return 0;

}

**Output (a):**

****

**Code (b):**

%{

#include <stdio.h>

#include <string.h>

// EOF Newline character is included by default

int spaces = 1, characters = 1;

int words = 0, lines = 0;

%}

%%

[ \t\n] {

spaces++;

characters++;

if (strcmp(yytext, "\n") == 0)

lines++;

}

[^ \t\n]+ {

if (yyleng > 1)

words++;

characters += yyleng;

}

%%

void result() {

printf("spaces:\t%d\n", spaces);

printf("words:\t%d\n", words);

printf("lines:\t%d\n", lines);

printf("characters:\t%d\n", characters);

}

void main()

{

yyin = fopen("test.c", "r");

yylex();

fclose(yyin);

result();

}

**test.c**

#include <stdio.h>

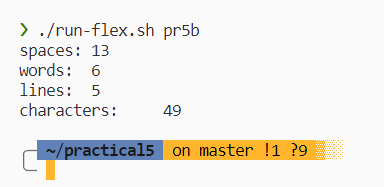
int main()

{

return 0;

}

**Output (b):**

****

**Code (c):**

%{

#include <stdio.h>

void write\_tokens(const char \*filename, char \*text);

%}

KEYWORDS auto|break|case|char|const|continue|default|do|double|else|enum|extern|float|for|goto|if|int|long|register|return|short|signed|sizeof|static|struct|switch|typedef|union|unsigned|void|volatile|while|printf|scanf|%d|include|stdio.h|main

OPERATORS []=|!|~|+|-|\*|/|%|^|\||&|<|>]

SEPARATORS ["{" | "}" | "(" | ")" | "\[" | "\]" | "." | "," | ";" | ":", "#", "\\", "\\n"]

IDENTIFIERS ([a-zA-Z][0-9])+|[a-zA-Z]\*

CONSTANTS [0-9]+

SPACES [ \t\n]+

%%

{SPACES} /\* Ignore \*/;

{CONSTANTS} /\* Ignore \*/;

{KEYWORDS}\* {

const char \*filename = "keywords.txt";

write\_tokens(filename, yytext);

}

{OPERATORS} {

const char \*filename = "operators.txt";

write\_tokens(filename, yytext);

}

{SEPARATORS} {

const char \*filename = "separators.txt";

write\_tokens(filename, yytext);

}

{IDENTIFIERS} {

const char \*filename = "identifiers.txt";

write\_tokens(filename, yytext);

}

%%

void write\_tokens(const char \*filename, char \*text) {

FILE \*f;

f = fopen(filename, "a");

fprintf(f, "%s ", text);

fclose(f);

}

void main()

{

yyin = fopen("test.c" , "r");

yylex();

fclose(yyin);

}

**test.c**

#include <stdio.h>

int main()

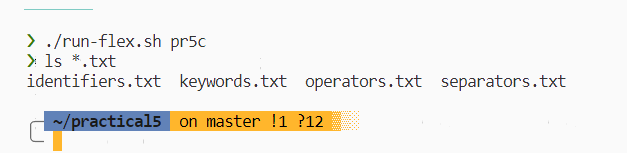
{

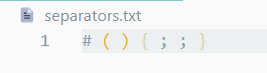
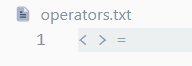
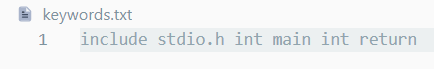
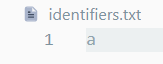
int a = 10;

return 0;

}

**Output(c):**

****

****