COMPILER DESIGN LAB

Week 4: Assignment

1. Implementation of lexical analyzer using LEX for recognizing the following tokens:

A:

```
%{
#include <stdio.h>
FILE *outputFile;
%}

%%
[/][*][^*]*[*]+([^*/][^*]*[*]+)*[/] {}
"#include<stdio.h>" { fprintf(outputFile, "This is Pre-processor directive: %s", yytext); }
"while"|"if"|"else" { fprintf(outputFile, "\nThis is keyword: %s", yytext); }
"int"|"float" { fprintf(outputFile, "\nThis is keyword: %s", yytext); }
"main()" { fprintf(outputFile, "\nThis is a function: %s", yytext); }
[a-zA-Z_][a-zA-Z0-9_]* { fprintf(outputFile, "\nThis is an identifier: %s", yytext); }
"<="|"=="|"="|"++"|"-"|"*"|"+"|"+="|"=+"|"-="|"=-" { fprintf(outputFile, "\nThis is an operator: %s",
yytext); }
")" {}
"["|"]" { fprintf(outputFile, "\nThis is a delimiter: %s", yytext); }
[(){}|,;] { fprintf(outputFile, "\nThis is a delimiter: %s", yytext); }
[\"\'] { fprintf(outputFile, "\nThis is a quote: %s", yytext); }
[0-9]*"."[0-9]+ { fprintf(outputFile, "\nThis is a float: %s", yytext); }
[0-9]+ { fprintf(outputFile, "\nThis is an integer: %s", yytext); }
. { fprintf(outputFile, "\nUnrecognized character: %s", yytext); }
%%

int yywrap()
{
    return 1;
}

int main()
{
    yyin = fopen("temp.txt", "r");
    outputFile = fopen("output.txt", "w");
    yylex();
    fclose(yyin);
    fclose(outputFile);
    return 0;
}
```

INPUT text:

```c
#include <stdio.h>

int main() {
    int x = 5;
    float y = 3.14;
    if (x <= 10) {
        y += 2.0;
    }
    return 0;
}
```

OUTPUT:

```
This is Pre-processor directive: #include<stdio.h>
This is keyword: int
This is a function: main()
This is an identifier: x
This is an operator: =
This is an integer: 5
This is a delimiter: ;
This is keyword: float
This is an identifier: y
This is an operator: =
This is a float: 3.14
This is keyword: if
This is a delimiter: (
This is an identifier: x
This is an operator: <=
This is an integer: 10
This is a delimiter: )
This is a delimiter: {
This is an identifier: y
This is an operator: +=
This is a float: 2.0
This is a delimiter: ;
This is a delimiter: }
```

2. Write a C Program to Scan and Count the number of characters, words, and lines in a file.

```
include <stdlib.h>

int main()
{
char fileName[100];

printf("Enter a filename: ");
scanf("%s", fileName);

FILE *file = fopen(fileName, "r");
int characterCount = 0;
int wordCount = 0;
int lineCount = 0;
int inWord = 0;
int c;

while ((c = fgetc(file)) != EOF)
{
characterCount++;
if (c == ' ' || c == '\t' || c == '\n' || c == '\r')
{
inWord = 0;
}
else if (!inWord)
{
inWord = 1;
wordCount++;
}
if (c == '\n')
{
lineCount++;
}
}
if (characterCount > 0 && c != '\n')
{
lineCount++;
}
printf("In the file %s:\n", fileName);
printf("Character count: %d\n", characterCount);
printf("Word count: %d\n", wordCount);
printf("Line count: %d\n", lineCount);
printf("\n-----------------------------");
fclose(file);
return 0;
}
```

INPUT text:

```
This is a sample text file.
It contains multiple lines of text.
Each line is counted as a line.
Words and characters are counted as well.
```

OUTPUT:

```
Enter a filename: sample.txt
In the file sample.txt:
Character count: 96
Word count: 19
Line count: 4

------------------------------
```


- **Submitted by**
  R. Jaya Rohith
  AP21110011323