

Lab Assignment 2

U21CS089 | Garvit Shah

1. Write a program to detect tokens in c program.

1. Input : A code file in C language
2. Output : list or table of different tokens.

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

bool isWhitespace(char c) {
    return (c == ' ' || c == '\t' || c == '\n' || c == '\r');
}

bool isOperator(char c) {
    return (c == '+' || c == '-' || c == '*' || c == '/' || c == '=' || c == '<'
|| c == '>');
}

bool isDelimiter(char c) {
    return (c == '(' || c == ')' || c == '{' || c == '}' || c == ',' || c ==
';');
}

bool isAlphanumeric(char c) {
    return (isalnum(c) || c == '_');
}

void processToken(const string& token) {
    cout << "Token: " << token << endl;
}

void tokenize(const string& code) {
    string token;
    bool inToken = false;

    for (char c : code) {
        if (isWhitespace(c) || isOperator(c) || isDelimiter(c)) {
            if (inToken) {
                processToken(token);
                token.clear();
                inToken = false;
            }
            if (!isWhitespace(c)) {
                processToken(string(1, c));
            }
        } else if (isAlphanumeric(c)) {
            token += c;
            inToken = true;
        }
    }

    if (inToken) {
        processToken(token);
    }
}

int main() {
```

```

    ifstream inputFile("test.cpp"); // Replace "input.cpp" with the actual
filename

    if (!inputFile.is_open()) {
        cout << "Error opening file." << endl;
        return 1;
    }

    string code((istreambuf_iterator<char>(inputFile)),
istreambuf_iterator<char>());
    inputFile.close();

    tokenize(code);

    return 0;
}

```

Test File:

```

#include <iostream>
using namespace std;

int main(){
    cout << "Hello";
    return 0;
}

```

```

>> ~/D/C/V/S/Lab cd "/Users/garvitshah/Desktop/Col
sktop/College/VI/SS/Lab/Assign2/"Q1
Token: include
Token: <
Token: iostream
Token: >
Token: using
Token: namespace
Token: std
Token: ;
Token: int
Token: main
Token: {
Token: }
Token: {
Token: cout
Token: <
Token: <
Token: Hello
Token: ;
Token: return
Token: 0
Token: ;
Token: }

```

2. Install lex tool in linux, and follow given manual to self-study. Write a Lex program to count the number of lines, characters and words of the given input file.

Lex File -

```
%{
#include<stdio.h>
int lc=0, wc=0, cc=0;
}%

%%
[\\n] {lc++; cc+=yylen; }
[^\\t\\n ]+ {wc++; cc+=yylen; }
%%

int main()
{
    printf("Enter the input:\\n");
    yylex();
    printf("No. of words are: %d\\n", wc);
    printf("No. of character are: %d\\n", cc);
    printf("No. of new lines are: %d\\n", lc);

    return 0;
}

int yywrap()
{
    return 1;
}
```

On Terminal -

```
> ~/D/C/V/S/L/Assign2 flex Q2.1
> ~/D/C/V/S/L/Assign2 gcc lex.yy.c -o q3_lexical_analyzer -ll
ld: warning: object file (/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/u
ld: warning: object file (/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/u
> ~/D/C/V/S/L/Assign2 ./lexer
Enter the input:
White crow and a purple unicorn world where reinvention is normalised

.
hey
No. of words are: 13
No. of character are: 67
No. of new lines are: 4
> ~/D/C/V/S/L/Assign2
```