

SS ASSIGNMENT – 6

Write a program to implement Lexical Analyzer(Lexer).

Program.txt:

```
int main()
{
    x = y+z;
    int x, y, z;
}
```

Source Code:

```
#include<iostream>
#include<fstream>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
using namespace std;
#define GREEN "\033[32m"
#define RESET "\033[0m"
#define BLUE "\033[34m"
#define MAGENTA "\033[35m"
#define CYAN "\033[36m"
#define YELLOW "\033[33m"

int isKeyword(char buffer[]){
    char keywords[32][10] = {"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"};

    int i, flag = 0;
    for(i = 0; i < 32; ++i){
        if(strcmp(keywords[i], buffer) == 0){
            flag = 1;
            break;
        }
    }
    return flag;
}

// Returns 'true' if the character is a DELIMITER.
bool isDelimiter(char ch)
```

```

{
    if (ch == ',' || ch == ';' || ch == '(' || ch == ')' ||
        ch == '[' || ch == ']' || ch == '{' || ch == '}')
        return (true);
    return (false);
}

// IS AN OPERATOR
bool isOperator(char ch)
{
    if (ch == '+' || ch == '-' || ch == '*' ||
        ch == '/' || ch == '>' || ch == '<' ||
        ch == '=')
        return (true);
    return (false);
}

// IS A VALID NUMBER
bool validNum(char ch)
{
    if (ch == '0' || ch == '1' || ch == '2' ||
        ch == '3' || ch == '4' || ch == '5' ||
        ch == '6' || ch == '7' || ch == '8' ||
        ch == '9')
        return (true);
    return (false);
}

//MAIN DRIVER
int main(){
    char ch, buffer[15];
    ifstream fin("program.txt");
    int i,j=0;
    if(!fin.is_open()){
        cout<<"error while opening the file\n";
        exit(0);
    }
    while(!fin.eof()){
        ch = fin.get();
        if(isalnum(ch)){
            buffer[j++] = ch;
        }else if((ch == ' ' || ch == '\n' || isOperator(ch) || isDelimiter(ch)) && (j != 0)){
            buffer[j] = '\0';
            j = 0;
            if(isKeyword(buffer) == 1)
                cout<<buffer<<" is "<<GREEN<<"KEYWORD"<<RESET<<endl;

            else{
                if (validNum(buffer[0])){
                    cout<<buffer<<" is "<<BLUE<<"INVALID IDENTIFIER"<<RESET<<endl;

```

```

        }else{
            cout<<buffer<<" is "<<CYAN<<"IDENTIFIER"<<RESET<<endl;
        }
    }
}
if(isOperator(ch))
    cout<<ch<<" is "<<MAGENTA<<"OPERATOR"<<RESET<<endl;
if(isDelimiter(ch))
    cout<<ch<<" is "<<YELLOW<<"SEPERATOR"<<RESET<<endl;
}
fin.close();
return 0;
}

```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS D:\assignments\SS\ASS06> cd "d:\assignments\SS\ASS06\" ; if ($?) { g++ q1.cpp -o q1 } ; if ($?) { .\q1 }
int is KEYWORD
main is IDENTIFIER
( is SEPERATOR
) is SEPERATOR
{ is SEPERATOR
x is IDENTIFIER
= is OPERATOR
y is IDENTIFIER
+ is OPERATOR
z is IDENTIFIER
; is SEPERATOR
int is KEYWORD
x is IDENTIFIER
, is SEPERATOR
y is IDENTIFIER
, is SEPERATOR
z is IDENTIFIER
; is SEPERATOR
} is SEPERATOR
PS D:\assignments\SS\ASS06>

```