Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

Lab Cycle 1 - Experiment 1

1. Write a program to design and implement a lexical analyzer using C language to recognize all valid tokens in the input program. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments.

Code:

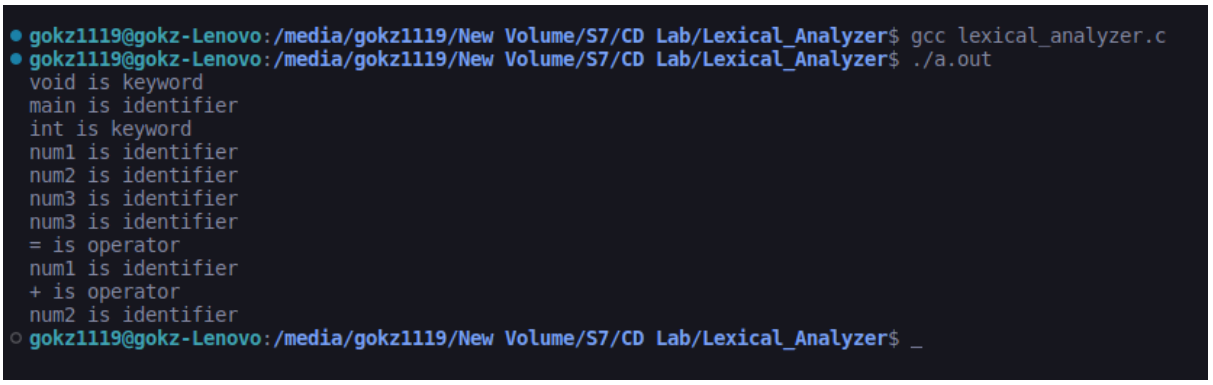
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

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

int main()
{
   char c, buffer[31], operators[] = "+-*/%=";
   FILE *fp;
   int i, j = 0;

   fp = fopen("Program.txt", "r");
```

```
    if (fp == NULL)
    {
        printf("Error while opening the file\n");
        exit(0);
    }

    while ((c = fgetc(fp)) != EOF)
    {
        for (i = 0; i < 6; ++i)
        {
            if (c == operators[i])
                printf("%c is operator\n", c);
        }
        if (isalnum(c))
        {
            buffer[j++] = c;
        }
        else if ((c == ' ' || c == '\t' || c == '\n') && (j != 0))
        {
            buffer[j] = '\0';
            j = 0;

            if (isKeyword(buffer) == 1)
                printf("%s is keyword\n", buffer);
            else
                printf("%s is identifier\n", buffer);
        }
    }
    fclose(fp);
    return 0;
}
```

Input File:

```
void main()
{
int num1, num2, num3;
num3 = num1 + num2;
}
```

Output:


gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Lexical_Analyzer$ gcc lexical_analyzer.c
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Lexical_Analyzer$ ./a.out
void is keyword
main is identifier
int is keyword
num1 is identifier
num2 is identifier
num3 is identifier
num3 is identifier
= is operator
num1 is identifier
+ is operator
num2 is identifier
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Lexical_Analyzer$ _