

Experiment No 2

Aim: Write a program to implement Lexical analyzer

Source Code:

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

bool ifKeyword(char word[]){
    if(
        !strcmp("auto",word) || !strcmp("double",word) ||
!strcmp("int",word) || !strcmp("struct",word) ||
        !strcmp("break",word) || !strcmp("else",word) ||
!strcmp("long",word) || !strcmp("switch",word) ||
        !strcmp("case",word) || !strcmp("enum",word) ||
!strcmp("register",word) || !strcmp("typedef",word) ||
        !strcmp("char",word) || !strcmp("extern",word) ||
!strcmp("return",word) || !strcmp("union",word) ||
        !strcmp("const",word) || !strcmp("float",word) ||
!strcmp("short",word) || !strcmp("unsigned",word) ||
        !strcmp("continue",word) || !strcmp("for",word) ||
!strcmp("signed",word) || !strcmp("void",word) ||
        !strcmp("default",word) || !strcmp("goto",word) ||
!strcmp("sizeof",word) || !strcmp("volatile",word) ||
        !strcmp("do",word) || !strcmp("if",word) || !strcmp("static",word)
|| !strcmp("while",word)
    ){
        return true;
    }
    return false;
}

bool ifOperator(char word[]){
    if(
        !strcmp("+",word) || !strcmp(">",word) || !strcmp("~",word) ||
!strcmp("%=",word) ||
        !strcmp("-",word) || !strcmp("<",word) || !strcmp("&",word) ||
!strcmp("<=",word) ||
        !strcmp("*",word) || !strcmp(">=",word) || !strcmp("^",word) ||
!strcmp(">>=",word) ||
        !strcmp("/",word) || !strcmp("<=",word) || !strcmp("|",word) ||
!strcmp("&=",word) ||
        !strcmp("%",word) || !strcmp("&&",word) || !strcmp("=",word) ||
!strcmp("^=",word) ||
        !strcmp("++",word) || !strcmp("||",word) || !strcmp("+=",word) ||
!strcmp("|=",word) ||
    )
```

```

        !strcmp("--",word) || !strcmp("!",word) || !strcmp("-=",word) ||
        !strcmp("==",word) || !strcmp("<<",word) || !strcmp("*=",word) ||
        !strcmp("!=" ,word) || !strcmp(">>",word) || !strcmp("/=",word)
    ){
        return true;
    }
    return false;
}

bool ifConstant(char word[]){
    for(int i=0;i<strlen(word);i++){
        if(word[i] == '.') continue;
        if(!isdigit(word[i])){
            return false;
        }
    }
    return true;
}

bool ifSpecialSymbol(char word[]){
    if(
        !strcmp("[",word) || !strcmp("]",word) || !strcmp("{",word) ||
!strcmp("}",word) ||
        !strcmp(",",word) || !strcmp(";",word) || !strcmp(":",word) ||
!strcmp("(",word) ||
        !strcmp(")",word)
    ){
        return true;
    }
    return false;
}

bool isLiteral(char word[]){
    int i = 0;
    int j = strlen(word) - 1;
    if(((int)word[i] == 34 && (int)word[i] == 34) || ((int)word[i] == 39
&& (int)word[j] == 39)){
        return true;
    }
    return false;
}

int main()
{
    char file_name[80] = "input.txt";
    FILE *in_file;
    char word[50];
    char c;

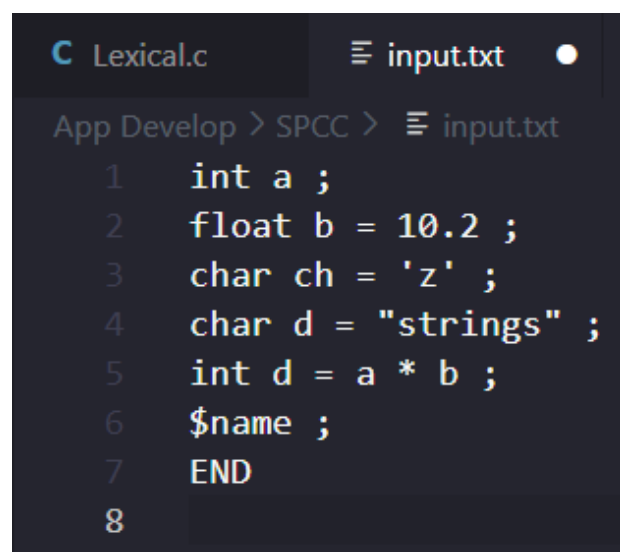
```

```

in_file = fopen(file_name, "r");
if (in_file == NULL)
    printf("Can't open %s for reading.\n", file_name);
else
{
    while (fscanf(in_file, "%s", word) != EOF)
    {
        if(ifKeyword(word) == true){
            printf("%s is a keyword\n",word);
        }
        else if(ifOperator(word) == true){
            printf("%s is a operator\n",word);
        }
        else if(ifConstant(word) == true){
            printf("%s is constant\n",word);
        }
        else if(ifSpecialSymbol(word) == true){
            printf("%s is special symbol\n",word);
        }
        else if(isLiteral(word) == true){
            printf("%s is a literal\n",word);
        }
        else{
            printf("%s is a identifier\n",word);
        }
    }
    fclose(in_file);
}
return 0;
}

```

Input File:



The screenshot shows a code editor with two tabs: 'Lexical.c' and 'input.txt'. The 'input.txt' tab is active, displaying the following C code:

```

1  int a ;
2  float b = 10.2 ;
3  char ch = 'z' ;
4  char d = "strings" ;
5  int d = a * b ;
6  $name ;
7  END
8

```

The editor's breadcrumb navigation shows 'App Develop > SPCC > input.txt'.

Output:

```
TERMINAL  ...  Code  +  -  [  [X]  ...  ^  X

PS C:\Users\kris\OneDrive\Documents\App Develop> cd
"c:\Users\kris\OneDrive\Documents\App Develop\SPCC\"
; if ($?) { gcc Lexical.c -o Lexical } ; if ($?) {
.\Lexical }
int is a keyword
a is a identifier
; is special symbol
float is a keyword
b is a identifier
= is a operator
10.2 is constant
; is special symbol
char is a keyword
ch is a identifier
= is a operator
'z' is a literal
; is special symbol
char is a keyword
d is a identifier
= is a operator
"strings" is a literal
; is special symbol
int is a keyword
d is a identifier
= is a operator
a is a identifier
* is a operator
b is a identifier
; is special symbol
$name is a identifier
; is special symbol
END is a identifier
PS C:\Users\kris\OneDrive\Documents\App Develop\SPCC
```