Bangladesh Army International University of Science and Technology

Department of Computer Science and Engineering

Lab Report

Lab Report No	:	8						
Lab Report Name	:	Write a Program to Detect Tokens in a CP Program						
Course Title	:	Compiler Design and Construction Sessional						
Course Code	:	CSE-414						
Name	:	Md Khaled Bin Joha						
ID	:	0822220105101052						
Level	:	4	**Term**	: 1	**Section**	: B	**Group**	:
Date of Submission	:	08/01/26	**Semester**	: FALL 25	**Year**	: 2026		

Key Learnings:

- **Lexical Analysis Fundamentals:** Learned how to break a source code file into **tokens** such as keywords, identifiers, numbers, operators, and special symbols.
- **File I/O in C:** Practiced reading source code from a file character by character using fgetc() and processing it programmatically.
- **Practical Compiler Insight:** Gained hands-on understanding of how **compilers or interpreters recognize language elements** before parsing.

Code Implementation:

Lab Works > lab8.c > main()

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

char keywords[10][10] = {
    "int", "float", "if", "else", "while",
    "for", "return", "char", "double", "void"
};

int isKeyword(char *word){
    for(int i=0; i<10; i++){
        if(strcmp(keywords[i], word) == 0){
            return 1;
        }
    }
    return 0;
}

int main(){
    char ch, buffer[50];
    int i=0;

    FILE *fp = fopen("input.c", "r");

    if(fp == NULL){
        printf("Error opening file\n");
        return 0;
    }

    printf("TOKENS FOUND\n");

    while((ch = fgetc(fp)) != EOF){
        // Preprocessor Symbol
        if(ch == '#'){
            printf("# -> Special symbol\n");

            fscanf(fp, "%s", buffer);
            printf("%s -> Preprocessor Directive\n", buffer);

```

Lab Works > lab8.c > main()

```c
            fgetc(fp);
            printf("< -> Special symbol\n");

            fscanf(fp, "%[^>]", buffer);
            printf("%s -> Header File\n", buffer);

            fgetc(fp);
            printf("> -> Special symbol\n");
        }
        else if(isalpha(ch)){
            buffer[i++] = ch;
            while(isalnum(ch = fgetc(fp))){
                buffer[i++] = ch;
            }

            buffer[i] = '\0';

            if(isKeyword(buffer)){
                printf("%s -> Keyword\n", buffer);
            }
            else{
                printf("%s -> Identifier\n", buffer);
            }

            i = 0;
            ungetc(ch, fp);
        }
        else if(isdigit(ch)){
            buffer[i++] = ch;
            while(isdigit(ch = fgetc(fp))){
                buffer[i++] = ch;
            }

            buffer[i] = '\0';
            printf("%s -> Number\n", buffer);

            i = 0;
            ungetc(ch, fp);
        }

```

```
79
80          else if(ch == '+' || ch == '-' ||  ch == '*' || ch == '/' || ch == '<' || ch == '>'){
81              printf("%c -> Operator\n", ch);
82          }
83
84          else if(ch == ';' || ch == ',' || ch == '(' || ch == ')' || ch == '{' || ch == '}'){
85              printf("%c -> Special Symbol\n", ch);
86          }
87      }
88      fclose(fp);
89      return 0;
90  }
```

Output Sample:

```
joha546@joha546:~/Projects/Compiler-Design-and-Construction/Lab Works$ ./lab8
TOKENS FOUND
# -> Special symbol
include -> Preprocessor Directive
< -> Special symbol
<stdio.h -> Header File
> -> Special symbol
int -> Keyword
main -> Identifier
( -> Special Symbol
) -> Special Symbol
{ -> Special Symbol
int -> Keyword
a -> Identifier
10 -> Number
; -> Special Symbol
float -> Keyword
b -> Identifier
a -> Identifier
+ -> Operator
5 -> Number
; -> Special Symbol
return -> Keyword
0 -> Number
; -> Special Symbol
} -> Special Symbol
```