**EXERCISE NO: 2** Date: 05 Feb, 2021

**Name- Devangi Purkayastha**

**Reg No- RA1811003010689**

**CSE – D2**

**Implementation of Lexical analyser for a C program**

**AIM:** To write a program for lexical analyser which takes a C file as the input file and converts the content as tokens.

**ALGORITHM:**

1. Read the C program file
2. Create lists of key, identifiers, format specifiers and io keywords
3. Read each line in the file, split the words in each line
4. If the word is in any of the above lists, append it to a separate list and repeat this step till the last line of the C program
5. Print the identifiers/tokens and their respective counts in the C program

**CODE:**

f=open(r"C:\Users\nikki\Desktop\study material\Compiler Design\Lab\ex2-LexicalAnalysis\ex2.c")

key=['int','float','string','include','stdio.h','char','break','if','else','switch','return','void'

,'while','struct','for']

iden=[]

sp={"(",")","{","}",";","&","#","$","\n",'"',","}

spec=["%d","%f","%c","%s"]

num="012345678910"

n=[]

k=[]

o=[]

l=[]

io=['scanf','printf','cin','cout']

op="+-%\*=/^><"

dl=[]

F=[]

for lines in f:

 words=lines.split(" ")

 for i in range(len(words)):

    if words[i] in key:

        k.append(words[i])

    elif words[i] in io:

        l.append(words[i])

    elif words[i] in op:

        o.append(words[i])

    elif words[i] in sp:

        dl.append(words[i])

    elif words[i] in spec:

        F.append(words[i])

    elif words[i] in num:

        n.append(words[i])

    else:

        iden.append(words[i])

print("Keywords are: ")

print(set(k))

print("input/output are: ")

print(set(l))

print("Operators are: ")

print(set(o))

print("Special Symbols are: ")

print(set(dl))

print("Identifiers are: " )

print(set(iden))

print("Format Specifier are:")

print(set(F))

print("Constants are:")

print(set(n))

**CODE IN C FILE –**

#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, flag = 0;

    for(i = 0; i < 32; ++i){

        if(strcmp(keywords[i], buffer) == 0){

            flag = 1;

            break;

        }

    }

    return flag;

}

int main(){

    char ch, buffer[15], operators[] = "+-\*/%=";

    FILE \*fp;

    int i,j=0;

    fp = fopen("C:/Users/nikki/Desktop/study material/Compiler Design/Lab/ex2-LexicalAnalysis/test.txt","r");

    if(fp == NULL){

        printf("error while opening the file\n");

        exit(0);

    }

    while((ch = fgetc(fp)) != EOF){

        for(i = 0; i < 6; ++i){

            if(ch == operators[i])

                printf("%c is operator\n", ch);

        }

        if(isalnum(ch)){

            buffer[j++] = ch;

        }

        else if((ch == ' ' || ch == '\n') && (j != 0)){

                buffer[j] = '\0';

                j = 0;

                if(isKeyword(buffer) == 1)

                    printf("%s is keyword\n", buffer);

                else

                    printf("%s is indentifier\n", buffer);

        }

    }

    fclose(fp);

    return 0;

}

**OUTPUT—**

****

**RESULT-**

The given program has been successfully executed.