**Ex.No: 2**

**LEXICAL ANALYZER USING C**

**Aim:**

To write a C program to implement Lexical Analyzer to identify the patterns like identifiers, keywords, comments, operators and constants.

**Description:**

The first step of a compiler is lexical analysis. It is the process of converting a sequence of characters into a sequence of tokens separated by white spaces and punctuation. A program that performs lexical analysis is calledlexical analyzer.

**Algorithm:**

1. Read the source program from the file.
2. Read and Separate a token using delimiter (whitespace or semicolon).
3. Check the token
   1. If the token starts with “\\” then display the entire line as COMMENTS
   2. If the token starts with “/\*” then display the lines until “\*/” as COMMENTS
   3. If the token starts with “#” then display that as PREPROCESSOR DIRECTIVE
   4. If the token is any keyword like int, if, else, etc.. display that as KEYWORD
   5. If the token is operator (+,-,\*, etc..) then display that as OPERATOR.
   6. If the token (,),{,} then display that as Special Symbol.
   7. If the token is numbers between 0 and 9 then display that as CONSTANT.
   8. Else check whether token starts with alphabet or ‘-‘. If so display that as identifier. Else print invalid identifier.
4. Write the output in a file.

**/\* Lexical Analyzer using C \*/**

#include<stdio.h>

void main()

{

char c,a[100];

FILE \*fptr;

fptr=fopen("sample.txt","r");

while(!feof(fptr))

{

fscanf(fptr,"%s",a);

if(!strcmp(a,"int")||!strcmp(a,"float"))

{

printf("\n%s-keyword",a);

fseek(fptr,1,1);

l:c=fgetc(fptr);

if(c==',')

{

printf("\n%c-others\n",c);

goto l;

}

if(c!=';')

{

if(c=='=')

{

printf("\n%c-operator",c);

c=fgetc(fptr);

printf("\n%c-constant",c);

}

printf("\n%c-identifier",c);

goto l;

}

}

else if(a[0]=='#')

printf("\n%s-preprocessor directive",a);

else if(!strcmp(a,"main()"))

printf("\n%s-identifier",a);

else

printf("\n%s-others\n",a);

}

fclose(fptr);

}

**/\*Input:Sample.txt\*/**

#include<stdio.h>

void main()

{

int a;

float b,c;

}

**/\*Output:\*/**

#include<stdio.h>-preprocessor directive

void-others

main()-identifier

{-others

int-keyword

a-identifier

float-keyword

b-identifier

,-others

c-identifier

}-others