LEXICAL ANALYSIS USING TRANSITION DIAGRAM

MOHAMMED FARHAN K N
S7 CSE B


```
#include<stdio.h>
void main()
{
	FILE *fi,*fo,*fop,*fk;
	int flag = 0,i = 1,flag1 = 0;
	char c,d,t,a[15],ch[20],file[20];
	fi = fopen("lexinput.c","r");
	fo = fopen("lexout.txt","w");
	fop = fopen("oper.txt","r");
	fk = fopen("key.txt","r");
	c = getc(fi);
	while(!feof(fi))
	{
		if(isalpha(c)||isdigit(c)||(c=='['||c==']'|| c=='.' ==1))
			fputc(c,fo);
		else if(c=='>'||c=='<' == 1)
		{
			d = getc(fi);
			if(d == '=')
				fprintf(fo,"\t%c%c\t",c,d);
			else
			{
				fseek(fi,-1,SEEK_CUR);
				fprintf(fo,"\t%c\t",c);
			}
		}

		else
		{
			if(c=='\n')
				fprintf(fo,"\t$\t");
			else
				fprintf(fo,"\t%c\t",c);
		}
		c = getc(fi);
	}
	fclose(fi);
	fclose(fo);
	fi = fopen("lexout.txt","r");
	//printf("\t\tLEXICAL ANALYSIS \n");
	fscanf(fi,"%s",a);
	printf("\nline : %d\n",i++);
	while(!feof(fi))
	{
		if((strcmp(a,"$")==0))
		{
			printf("\nline : %d\n",i++);
			fscanf(fi,"%s",a);
		}
		fscanf(fop,"%s",ch);
		while(!feof(fop))
		{
			if(strcmp(ch,a)==0)
```

```
                {
                        fscanf(fop,"%s",ch);
                        printf("\t%s\t--->\t%s\n",a,ch);
                        flag=1;
                }
                fscanf(fop,"%s",ch);
        }
        rewind(fop);
        fscanf(fk,"%s",ch);
        while(!feof(fk))
        {
                if(strcmp(ch,a)==0)
                {
                        fscanf(fk,"%s",ch);
                        printf("\t%s\t--->\tKeyword\n",a);
                        flag=1;
                }
                fscanf(fk,"%s",ch);
        }
        rewind(fk);
        if(flag==0)
        {
                if(isdigit(a[0]))
                        printf("\t%s\t--->\tNumber\n",a);
                else
                        printf("\t%s\t--->\tIdentifier\n",a);

        }
        flag=0;
        fscanf(fi,"%s",a);
    }
    fclose(fi);
    fclose(fk);
    fclose(fop);
}
```

INPUT_FILE

```
#include<stdio.h>
main()
{
    int a5,a[5],b,c;
    a = 25;
    b = 13;
    c = a - b;
    if(a<=b)
     b=5;
    printf("%d\n ",c);
}
```

INTERMEDIATE FILE

```
#  include  <  stdio.h  >  $  main  (  )  $ { $
   int  a5  ,  a[5]  ,  b ,  c  ; $
a  = 25  ;  $ b  = 13  ;  $
   c = a  -  b  ;  $ if  (
a  <= b  )  $  b  =  5
; $  printf  (  "  %  d  \  n
   "  ,  c  )  ;  $  }
```

$

OPERATOR FILE

(OpenParanthesis
) CloseParanthesis
{ Openbrace
} Closingbrace
< LT
> GT
<= LE
>= GE
" DoubleQuote
' SingleQuote
: Colon
; Semicolon
Preprocessor
= Assign
== Equal
% Percentage
^ Bitwise
& Reference
* Asterics
+ Addition
- Subtraction
\ Backslash
/ Forwardslash
, Comma
%f Float
\n Newline

KEYWORD FILE

int
void
main
char
if
for
while
else
printf
scanf
FILE
include
stdio.h
conio.h
iostream.h
float

OUTPUT

```
line : 1
        #       --->    Preprocessor
        include --->    Keyword
        <       --->    LT
        stdio.h --->    Keyword
        >       --->    GT

line : 2
        main    --->    Keyword
        (       --->    OpenParanthesis
        )       --->    CloseParanthesis

line : 3
        {       --->    Openbrace

line : 4
        int     --->    Keyword
        a5      --->    Identifier
        ,       --->    Comma
        a[5]    --->    Identifier
        ,       --->    Comma
        b       --->    Identifier
        ,       --->    Comma
        c       --->    Identifier
        ;       --->    Semicolon

line : 5
        a       --->    Identifier
        =       --->    Assign
        25      --->    Number
        ;       --->    Semicolon

line : 6
        b       --->    Identifier
        =       --->    Assign
        13      --->    Number
        ;       --->    Semicolon

line : 7
        c       --->    Identifier
        =       --->    Assign
        a       --->    Identifier
        -       --->    Subtraction
        b       --->    Identifier
        ;       --->    Semicolon

line : 8
        if      --->    Keyword
        (       --->    OpenParanthesis
        a       --->    Identifier
        <=      --->    LE
        b       --->    Identifier
        )       --->    CloseParanthesis

line : 9
        b       --->    Identifier
```

```
        =       --->    Assign
        5       --->    Number
        ;       --->    Semicolon

line : 10
        printf  --->    Keyword
        (       --->    OpenParanthesis
        "       --->    DoubleQuote
        %       --->    Percentage
        d       --->    Identifier
        \       --->    Backslash
        n       --->    Identifier
        "       --->    DoubleQuote
        ,       --->    Comma
        c       --->    Identifier
        )       --->    CloseParanthesis
        ;       --->    Semicolon

line : 11
        }       --->    Closingbrace
```