

Practical-3

AIM : Write a C program to develop a lexical analyzer to recognize a few tokens in C. (Note: Read the small C program from file and recognize a tokens like Identifiers, Operators, Comments, Constants, Special Symbols etc.)

Input: Read the program from the file:

Input.c

```
int a = 5, b = 6,c;
c= a + b;
```

Expected output:

Symbol	Token
int	Keyword
a	Identifier
=	Assignment operator
5	Constant
b	Identifier
=	Assignment operator
6	Constant
c	Identifier

- Code :**

```
#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[] = "+-*/%<=>==";
    char pf[]="printf";
    char sf[]="scanf";
    char str[]=" \" \" ";
    FILE *fp;
    int i,j=0;

    fp = fopen("input.c", "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==" ")
            {
                printf(" ");
            }
            if(ch == operators[i])
                printf("%c\t:\tis operator\n", ch);
        }
        if(ch==pf[i])
        {
            printf("Printf statement");

```

```

        }
        if(ch==sf[i])
        {
printf("Scanf statement");

        }
        if(ch==str[i])
        {
printf("Scanf statement");

        }

    if(isdigit(ch)){
        printf("%c\t:tis Constant\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\t:tis keyword\n", buffer);
        else
            printf("%s\t:tis indentifier\n", buffer);
    }
}

fclose(fp);
FILE *filePointer;
char chrr;
int x=0,y=1,flg=0;

// char *strr=" ";

//2
filePointer = fopen("input.c", "r");

```

```
//3
if (filePointer == NULL)
{
    printf("File is not available \n");
}
else
{
    //4
    while ((chrr = fgetc(filePointer)) != EOF)
    {
        if(x==1)
        {

            if(chrr=="")
            { x=0;
              printf(" : string-%d\n",y);
              y++;
              continue; }

            printf("%c",chrr);
        }

        if(chrr=="")
        {
            x=1;
            continue;
        }

        if(chrr=="")
            flg=1;
        if(chrr=",")
            flg=1;

    }

    if(flg==1)
    {

        printf("; : special characters \n, : special characters");
    }

}

//5
fclose(filePointer);

return 0;
```

```
}
```

- **INPUT.C :-**

```
void main()
{
    int a,b,c=5;
    printf("CD PR 3 ");
    c=a*b;
    for(i=0;i<5;i++)
    {
        if(i==4)
        {
            break;
        }
        if(i==3)
        {
            continue;
        }
        printf("Kishan Patel");
        printf(i);
    }
}
```

- **OUTPUT :-**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Dell Store> cd "c:\TURBOC3\kishan"
PS C:\TURBOC3\kishan> cmd /c .\"cdpr3.exe"
void      :      is keyword
main      :      is indentifier
int       :      is keyword
5         :      is Constant
abc5      :      is indentifier
printfCD  :      is indentifier
PR        :      is indentifier
3         :      is Constant
3         :      is indentifier
*         :      is operator
cab       :      is indentifier
0         :      is Constant
<         :      is operator
5         :      is Constant
+         :      is operator
+         :      is operator
```

```
fori0i5i  :      is indentifier
4         :      is Constant
ifi4      :      is indentifier
break     :      is keyword
3         :      is Constant
ifi3      :      is indentifier
continue  :      is keyword
printfKishan :      is indentifier
Patel     :      is indentifier
printfi   :      is indentifier
CD PR 3   : string-1
Kishan Patel : string-2
; : special characters
, : special characters
PS C:\TURBOC3\kishan> 
```