Practical no 2

**Program :**

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

void main()

{

char str[30];

int state=0,i=0;

//clrscr();

printf("\n\t\*\*\*\*\*\* LEXICAL ANALYZER \*\*\*\*\*\*\n\t");

printf("\n\n\tPlease enter any string:\n\t");

gets(str);

while(str[i]!=NULL)

{

switch(state)

{

case 0: if(str[i]=='e')

{

state=1;

}

else if(str[i]=='i')

{

state=4;

}

else if(str[i]=='f')

{

state=5;

}

else if(str[i]=='d')

{

state=7;

}

else if(str[i]=='w')

{

state=8;

}

else if(isalpha(str[i]))

{

state=12;

}

else if(isdigit(str[i])||(str[i]=='.'))

{

state=13;

}

else

{

state=0;

i++;

}

break;

case 1: i++;

if(str[i]=='l')

{

state=2;

}

else

{

state=12;

i=i-1;

}

break;

case 2: i++;

if(str[i]=='s')

{

state=3;

}

else

{

state=12;

i=i-2;

}

break;

case 3: i++;

if(str[i]=='e')

{

printf("\n else is a keyword");

state=0;

i++;

}

else

{

state=12;

i=i-3;

}

break;

case 4: i++;

if(str[i]=='f')

{

printf("\n if is a keyword");

state=0;

i++;

}

else

{

state=12;

i=i-1;

}

break;

case 5: i++;

if(str[i]=='o')

{

state=6;

}

else

{

state=12;

i=i-1;

}

break;

case 6:i++;

if(str[i]=='r')

{

printf("\n for is a keyword");

state=0;

i++;

}

else

{

state=12;

i=i-2;

}

break;

case 7:i++;

if(str[i]=='o')

{

printf("\n do is a keyword");

state=0;

i++;

}

else

{

state=12;

i=i-1;

}

break;

case 8:i++;

if(str[i]=='h')

{

state=9;

}

else

{

state=12;

i=i-1;

}

break;

case 9:i++;

if(str[i]=='i')

{

state=10;

}

else

{

state=12;

i=i-2;

}

break;

case 10:i++;

if(str[i]=='l')

{

state=11;

}

else

{

state=12;

i=i-3;

}

break;

case 11:i++;

if(str[i]=='e')

{

printf("\n while is a keyword");

state=0;

i++;

}

else

{

state=12;

i=i-4;

}

break;

case 12:printf("\n\t%c",str[i]);

while(isalnum(str[++i]))

{

printf("%c",str[i]);

}

printf(" :is an identifier");

state=0;

break;

case 13:printf("\n\t%c",str[i]);

while(isdigit(str[++i])||(str[i]=='.'))

{

printf("%c",str[i]);

}

printf(" is a constant\n");

state=0;

break;

}

}

getch();

}

**Output :**

