**CD Practical 4:**

**Write a C program to simulate lexical analyser for validating operators.**

#include<stdio.h>

#include<conio.h>

#include <string.h>

void main()

{

char s[5];

printf("\n Enter any operator:");

gets(s);

switch(s[0])

{

case'>': if(s[1]!='>' && strlen(s)==1) printf("\n Greater than \n");

else if(s[0]=='>' && s[1]=='=') printf("\n Greater than or equal\n");

else if(s[0]=='>' && s[1]=='>') printf("\n Right shift\n");

else printf("\n There is no such operator\n");

break;

case'<': if(s[1]!='<' && strlen(s)==1) printf("\n Less than \n");

else if(s[0]=='<' && s[1]=='=') printf("\n Less than or equal\n");

else if(s[0]=='<' && s[1]=='<') printf("\n Left shift\n");

else printf("\n There is no such operator\n");

break;

case'=': if(s[1]!='=' && strlen(s)==1) printf("\n Assignment operator \n");

else if(s[0]=='=' && s[1]=='=' && strlen(s)==2) printf("\n equals to \n");

else printf("\n There is no such operator\n");

break;

case'!': if(s[1]!='!' && strlen(s)==1) printf("\nBitwise NOT\n");

else if(s[0]=='!' && s[1]=='=' && strlen(s)==2) printf("\n not equals \n");

else printf("\n There is no such operator\n");

break;

case'&': if(s[1]!='&' && strlen(s)==1) printf("\nBitwise AND\n");

else if(s[0]=='&' && s[1]=='&' && strlen(s)==2) printf("\nLogical AND\n");

else printf("\n There is no such operator\n");

break;

case'|': if(s[1]!='|' && strlen(s)==1) printf("\nBitwise OR\n");

else if(s[0]=='|' && s[1]=='|') printf("\nLogical OR\n");

else printf("\n There is no such operator\n");

break;

case'+': if(s[1]!='+' && strlen(s)==1) printf("\n Addition\n");

else if(s[0]=='+' && s[1]=='+' && strlen(s)==2) printf("\nIncrement operator\n");

else printf("\n There is no such operator\n");

break;

case'-': if(s[1]!='-' && strlen(s)==1) printf("\n Subtraction\n");

else if(s[0]=='-' && s[1]=='-' && strlen(s)==2) printf("\nDecrement operator\n");

else printf("\n There is no such operator\n");

break;

case'\*': if(s[1]!='\*' && strlen(s)==1) printf("\n Multiplication operator \n");

else printf("\n There is no such operator\n");

break;

case'/': if(s[1]!='/' && strlen(s)==1) printf("\n Division operator \n");

else printf("\n There is no such operator\n");

break;

case'%': if(s[1]!='%' && strlen(s)==1) printf("\n Modulus operator \n");

else printf("\n There is no such operator\n");

break;

case'^': if(s[1]!='^' && strlen(s)==1) printf("\n Bitwise exclusive OR operator \n");

else printf("\n There is no such operator\n");

break;

default: printf("\n There is no such operator");

}

}

**Program output:**





