# Practical No. 5

Aim : To perform c program for Scanner.

Program :

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
	char str[30];
	int state=0,i=0;
	printf("\n\t****** LEXICAL ANALYZER ******\n\t");
	printf("\n\n\tPlease enter any string:\n\t");
	scanf("%s",str);
	while(str[i]!='\0')
	{
		while(str[i]==' ')
		{
			i++;
		}
		switch(state)
		{
		case 0: if(str[i]=='e')
			{
				state=1;
			}
			else if(str[i]=='i')
			{
				state=6;
			}
			else if(str[i]=='f')
```

```
{
        state=9;
}
else if(str[i]=='d')
{
        state=13;
}
else if(str[i]=='w')
{
        state=16;
}
else if(isalpha(str[i]))
{
        state=22;
}
else if(isdigit(str[i]))
{
        state=23;
}
else if(str[i]=='+')
{
        state=24;
}
else if(str[i]=='-')
{
        state=26;
}
else if(str[i]=='=')
{
        state=28;
}
```

```
else if(str[i]=='<')
{
        state=30;
}
else if(str[i]=='>')
{
        state=32;
}
else if(str[i]=='}')
{
        state=34;
}
else if(str[i]=='(')
{
        state=35;
}
else if(str[i]==')')
{
        state=36;
}
else if(str[i]=='[')
        {
        state=37;
}
else if(str[i]==']')
{
        state=38;
}
else if(str[i]==';')
{
        state=39;
```

```
		}
		else if(str[i]=='/')
		{
			state=40;
		}
		else if(str[i]=='*')
		{
			state=41;
		}
		else if(str[i]=='{')
		{
			state=42;
		}
		else
		{
			state=22;
		}
		break;

case 1: i++;
		if(str[i]=='l')
		{
			state=2;
		}
		else
		{
			state=2;
			i=i-1;
		}
		break;
case 2: i++;
```

```
            if(str[i]=='s')
            {
                        state=3;
            }
            else
            {
                        state=22;
                        i=i-2;
            }
            break;
case 3: i++;
            if(str[i]=='e')
            {
                        state=4;
            }
            else
            {
                        state=22;
                        i=i-3;
            }
            break;
case 4: i++;
            if(str[i]=='{')
            {
                        state=5;
            }
            else
            {
                        state=22;
                        i=i-4;
            }
```

```
            break;
case 5: i++;
            printf("\n else: is a keyword");
            printf("\n {: is a punctuation");
            state=0;
            i++;
            break;
case 6: i++;
            if(str[i]=='f')
            {
                        state=7;
            }
            else
            {
                        state=22;
                        i=i-1;
            }
            break;
case 7: i++;
            if(str[i]=='(')
            {
                        state=8;
            }
            else
            {
                        state=22;
                        i=i-2;
            }
            break;
case 8: printf("\n if: is a keyword");
            printf("\n (: is a punctuation");
```

```
                state=0;
                i++;
                break;
case 9: i++;
                if(str[i]=='o')
                {
                        state=10;
                }
                else
                {
                        state=22;
                        i=i-1;
                }
                break;
case 10:i++;
                if(str[i]=='r')
                {
                        state=11;
                }
                else
                {
                        state=22;
                        i=i-2;
                }
                break;
case 11:i++;
                if(str[i]=='(')
                {
                        state=12;
                }
                else
```

```
        {
                state=22;
                i=i-3;
        }
        break;

case 12:printf("\n for: is a keyword");
        printf("\n (: is a punctuation");
        state=0;
        i++;
        break;
case 13:i++;
        if(str[i]=='0')
        {
                state=14;
        }
        else
        {
                state=22;
                i=i-1;
        }
        break;
case 14:i++;
        if(str[i]=='{')
        {
                state=15;
        }
        else
        {
                state=22;
                i=i-2;
```

```
            }
            break;
case 15:printf("\n do: is a keyword");
            printf("\n {: is a punctuation");
            state=0;
            i++;
            break;
case 16:i++;
            if(str[i]=='h')
            {
                        state=17;
            }
            else
            {
                        state=22;
                        i=i-1;
            }
            break;
case 17:i++;
            if(str[i]=='i')
            {
                        state=18;
            }
            else
            {
                        state=22;
                        i=i-2;
            }
            break;
case 18:i++;
            if(str[i]=='l')
```

```
		{
			state=19;
		}
		else
		{
			state=22;
			i=i-3;
		}
		break;
case 19:i++;
		if(str[i]=='e')
		{
			state=20;
		}
		else
		{
			state=22;
			i=i-4;
		}
		break;
case 20:i++;
		if(str[i]=='(')
		{
			state=21;
		}
		else
		{
			state=22;
			i=i-5;
		}
		break;
```

```
case 21:printf("\n while: is a keyword");
        printf("\n (: is a punctuation");
        state=0;
        i++;
        break;
case 22:printf("\n\t%c", str[i]);
        while(isalnum(str[++i]))
        {
                printf("%c", str[i]);
        }
        printf(": is an identifier");
        state=0;
        break;
case 23:printf("\n\t%c", str[i]);
        while(isdigit(str[++i]))
        {
                printf("%c", str[i]);
        }
        printf(": is a constant");
        state=0;
        break;
case 24:i++;
        if(str[i]=='+')
        {
                state=25;
        }
        else
        {
                i--;
                printf("\n\t +: is an arithmatic operator");
                state=0;
```

```
                i=i+1;
        }
        break;
case 25:printf("\n\t+: is an increment operator");
        state=0;
        i++;
        break;
case 26:i++;
        if(str[i]=='-')
        {
                state=27;
        }
        else
        {
                i--;
                printf("\n\t-: is an arithmatic operator");
                state=0;
                i=i+1;
        }
        break;
case 27:printf("\n\t-: is a decrement operator");
        state=0;
        i++;
        break;
case 28:i++;
        if(str[i]=='=')
        {
                state=29;
        }
        else
        {
```

```
                i--;
                printf("\n\t=: is an assignment operator");
                state=0;
                i=i+1;
        }
        break;
case 29:printf("\n\t=: is a relational operator");
        state=0;
        i++;
        break;
case 30:i++;
        if(str[i]=='<')
        {
                state=31;
        }
        else
        {
                i--;
                printf("\n\t<: is a relational operator");
                state=0;
                i=i+1;
        }
        break;
case 31:printf("\n\t<: is a relational operator");
        state=0;
        i++;
        break;
case 32:i++;
        if(str[i]=='>')
        {
                state=33;
```

```
		}
		else
		{
			i--;
			printf("\n\t>: is a relational operator");
			state=0;
			i=i+1;
		}
		break;
case 33:printf("\n\t>: is a relationl operator");
		state=0;
		i++;
		break;
case 34:printf("\n\t}: is a punctuation");
		state=0;
		i++;
		break;
case 35:printf("\n\t(: is a punctuation");
		state=0;
		i++;
		break;
case 36:printf("\n\t): is a punctuation");
		state=0;
		i++;
		break;
case 37:printf("\n\t[: is a punctuation");
		state=0;
		i++;
		break;
case 38:printf("\n\t]: is a punctuation");
		state=0;
```

```
                    i++;
                    break;
                case 39:printf("\n\t;: is a semicolon");
                    state=0;
                    i++;
                    break;
                case 40:printf("\n\t/: is an arithmatic operator");
                    state=0;
                    i++;
                    break;
                case 41:printf("\n\t+: is an arithmatic operator");
                    state=0;
                    i++;
                    break;
                case 42:printf("\n\t}: is a punctuation");
                    state=0;
                    i++;
                    break;
                }
        }
getch();
}
```

```
        ****** LEXICAL ANALYZER ******


        Please enter any string:
        for()

for: is a keyword
(: is a punctuation
        ): is a punctuation
```

```
C:\Users\prite\AppData\Local

        ****** LEXICAL ANALYZER ******


        Please enter any string:
        13579

        13579: is a constant
```

```
C:\Users\prite\AppData\Local

        ****** LEXICAL ANALYZER ******


        Please enter any string:
        +

         +: is an arithmatic operator
--------------------------------
Process exited after 13.8 seconds with return value 13
Press any key to continue . . .
```