Ex. No: 1

Date:

Implementation of Lexical Analyzer for 'if' Statement

Aim:

To write a C program to implement lexical analyzer for 'if' statement.

Algorithm:

1. Input: Programming language 'if' statement Output: A sequence of tokens.
2. Tokens have to be iden its respective attributes have to be printed.

Lexeme	**Token**
********	*******
If	<1,1>
variable-name	<2,#address>
numeric-constant	<3,#address>
;	<4,4>
(	<5,0>
)	<5,1>
{	<6,0>
}	<6,1>
>	<62,62>
>=	<620,620>
<	<60,60>
<=	<600,600>
!	<33,33>
!=	<330,330>
=	<61,61>
==	<610,610>

Program:

```
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<string.h>
char vars[100][100];
int vcnt;
char input[1000],c;
char token[50],tlen;
int state=0,pos=0,i=0,id;
```

```
char*getAddress(char str[])
{
for(i=0;i<vcnt;i++)
if(strcmp(str,vars[i])==0)
return vars[i];
strcpy(vars[vcnt],str);
return vars[vcnt++];
}
intisrelop(char c)
{
if(c=='>'||c=='<'||c=='|'||c=='=')
return 1;
else
return 0;
}
int main(void)
{
clrscr();
printf("Enter the Input String:");
gets(input);
do
{
c=input[pos];
putchar(c);
switch(state)
{
case 0:
if(c=='i')
state=1;
break;
case 1:
if(c=='f'
)
{
printf("\t<1,1>\n");
state =2;
}
break;
case 2:
if(isspace(c))
printf("\b");
if(isalpha(c))
{
token[0]=c;
tlen=1;
state=3;
}
if(isdigit(c))
state=4;
```

```
if(isrelop(c))
state=5;
if(c==';')printf("\t<4,4>\n");
if(c=='(')printf("\t<5,0>\n");
if(c==')')printf("\t<5,1>\n");
if(c=='{') printf("\t<6,1>\n");
if(c=='}') printf("\t<6,2>\n");
break;
case 3:
 if(!isalnum(c))
{
token[tlen]='\0';
printf("\b\t<2,%p>\n",getAddress(token));
state=2;
pos--;
}
else
token[tlen++]=c;
break;
case 4:
if(!isdigit(c))
{
printf("\b\t<3,%p>\n",&input[pos]);
state=2;
pos--;
}
break;
case 5:
id=input[pos-1];
if(c=='=')
printf("\t<%d,%d>\n",id*10,id*10);
else
{
printf("\b\t<%d,%d>\n",id,id);
pos--;
}
state=2;
break;
}
pos++;
}
while(c!=0);
getch();
return 0;
}
```

Sample Input & Output:

Enter the input string: if(a>=b) max=a;

if	<1,1>
(	<5,0>
a	<2,0960>
>=	<620,620>
b	<2,09c4>
)	<5,1>
max	<2,0A28>
=	<61,61>
a	<2,0A8c>
;	<4,4>

Result:

The above C program was successfully executed and verified.