Name: Bhavya Patel

Roll No: 20BCE198

Course: Compiler construction

Practical No: 1

Aim: To implement lexical analyse to recognize all distinct token classes: use flex/lex tool to recognize all distinct token classes (Data type, Identifier, constant (Integer, Float, Char, String), Operator (Arithmetic, Relational, Assign, Unary +/-, Increment), Single line/Multi-line comments, Special symbol(;,{}())) . Generate Lexical error reports for invalid lexeme.

Methodology followed:

```
%{
#include <stdio.h>
%}
%%
(["]|['])[a-zA-z0-9](["]|[']) {printf("char const ");}
(["]|['']).*(["]|[']) {printf("string ");}
"/*"([^*]|\n|[*][^/])*"*/" {printf("multiline comment ");}
"//"[a-zA-Z0-9 ]* {printf("singleline comment ");}
[0-9]*[.][0-9]* {printf("float value ");}
[0-9]* {printf("integer value ");}
"+"|"-"|"*"|"/"|"%"|"++"|"-"|"<="|">=" {printf("special symbol ");}
";" {printf("\n");}
[ ] {printf(" ");}
"printf" {printf("printf ");}
"if"|"else"|"for"|"while"|"break"|"continue" {printf("keyword ");}
"int"|"float"|"double"|"char" {printf("datatype ");}
[a-zA-Z_][a-zA-Z0-9_]* {printf("Identifier ");}
[.]* {printf("%s",yytext);}
%%
main()
```

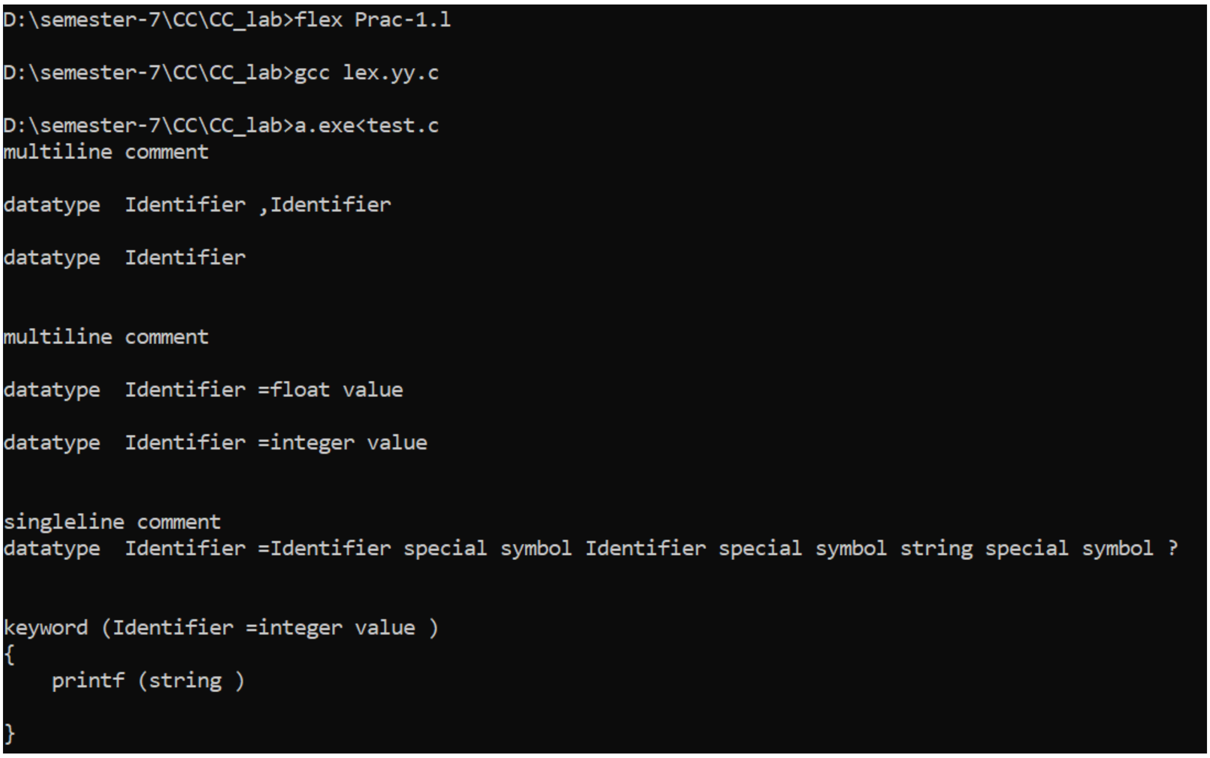
```
{
yylex();
}
int yywrap()
{
    return 0;
}
```

Test file:

Test.c

```
/* najbdacha


hello 123

/*
abjacb


 */

int k123,k1234;
float a123;

/*

hello 1234
hello 123
*/

int abcd12=32.56;
int pqr123_=32;

//hello 123
int a=b+c+'c'+'?'+?;

if(a=1)
{
    printf("hello   ???  2424");
}
```

Output:



```
D:\semester-7\CC\CC_lab>flex Prac-1.l

D:\semester-7\CC\CC_lab>gcc lex.yy.c

D:\semester-7\CC\CC_lab>a.exe<test.c
multiline comment

datatype  Identifier ,Identifier

datatype  Identifier


multiline comment

datatype  Identifier =float value

datatype  Identifier =integer value


singleline comment
datatype  Identifier =Identifier special symbol Identifier special symbol string special symbol ?


keyword (Identifier =integer value )
{
    printf (string )

}
```

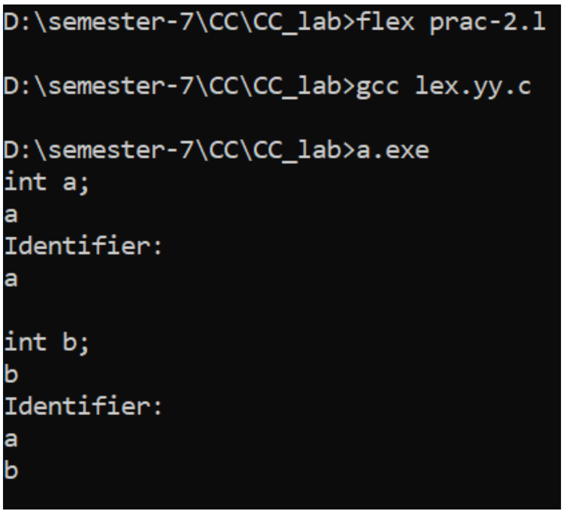
AIM: Implement symbol table

Methodology followed:

```
%{

#include <stdio.h>

#include<string.h>

char symbols[10][20];

int ind=0;

void symadd(char name[],int n){

    int size= sizeof(name) / sizeof(name[0]);

    int i=0,p=0;

    for(;i<n;i++){

    }

    while(i<size && name[i]==' ')i++;

    while(name[i]!=';'){
        symbols[ind][p]=name[i];
        printf("%c\n",symbols[ind][p]);
        p++;i++;
    }
    symbols[ind][p]='#';
    ind++;
}
void print(){
    printf("Identifier:\n");
    int i,j;
    //printf("%d",ind);
    for(i=0;i<ind;i++){
        int n=sizeof(symbols[i]) / sizeof(symbols[i][0]);
        for(j=0;j<n;j++){
            if(symbols[i][j]!='#')
            printf("%c\n",symbols[i][j]);
            else break;
        }
    }
}
%}
%%
"int"[ ]*[a-zA-Z_][a-zA-Z0-9_]*";" {symadd(yytext,3);print();}
"float"[ ]*[a-zA-Z_][a-zA-Z0-9_]*";" {symadd(yytext,4);print();}
"double"[ ]*[a-zA-Z_][a-zA-Z0-9_]*";" {symadd(yytext,5);print();}
"char"[ ]*[a-zA-Z_][a-zA-Z0-9_]*";" {symadd(yytext,4);print();}
%%

main()
{
```

```
yylex();
}
int yywrap()
{
    return 0;
}
```

Output:



```
D:\semester-7\CC\CC_lab>flex prac-2.l

D:\semester-7\CC\CC_lab>gcc lex.yy.c

D:\semester-7\CC\CC_lab>a.exe
int a;
a
Identifier:
a

int b;
b
Identifier:
a
b
```