NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

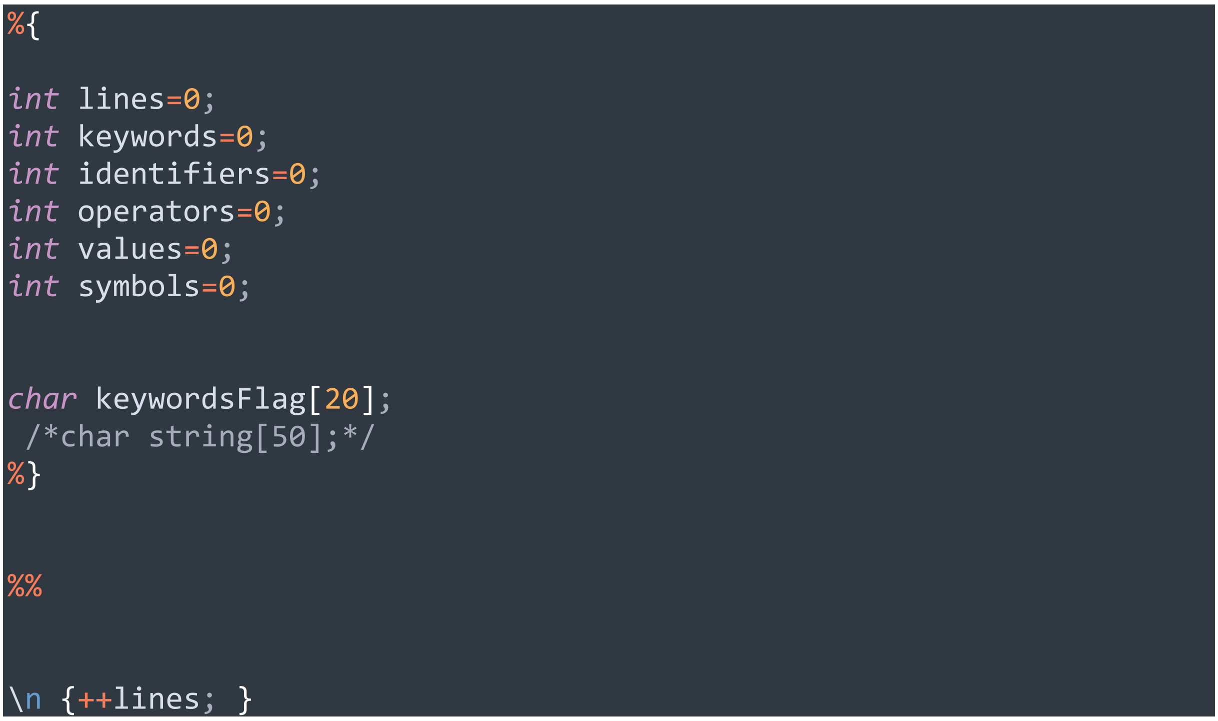
Name: PRATYUSH V KHARE **Subject: Compiler Const.**

Roll No: 20BCE519 **Semester :- 7** **Subject Code: 2CS701**

Practical 1

Aim: To implement lexical analyser 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(;,{}())) .

P1.l Code



```
%{

int lines=0;
int keywords=0;
int identifiers=0;
int operators=0;
int values=0;
int symbols=0;


char keywordsFlag[20];
 /*char string[50];*/
%}


%%


\n {++lines; }
```

```
"int"|"float"|"while"|"if"|"else"|"for"|"do"|"void"  {keywords++;
printf("\nLine => %d Keywords = %s",lines,yytext);}
"<="|"=="|"="|"++"|"-"|"*"|"+" {operators++; printf("\nLine => %d
Operators = %s",lines,yytext);}
[(){}|, ;] {symbols++; printf("\nLine => %d Symbols =
%s",lines,yytext);}
[0-9]*"."[0-9]+ {values++; printf("\nLine => %d Values =
%s",lines,yytext);}
[0-9]+ {values++; printf("\nLine => %d Values = %s",lines,yytext);}
[a-zA-Z_][a-zA-Z0-9_]* {identifiers++; printf("\nLine => %d
Identifiers = %s",lines,yytext);}


%%

int main(){
      yylex();
      printf("\n\n \t-------------OUTPUT------------- ");
      printf("\n Total number of keywords = %d\n", keywords);
      printf("\n Total number of identifiers = %d\n", identifiers);
      printf("\n Total number of operators = %d\n", operators);
      printf("\n Total number of values = %d\n", values);
      printf("\n Total number of symbols = %d\n", symbols);

}

int yywrap()
{}
```

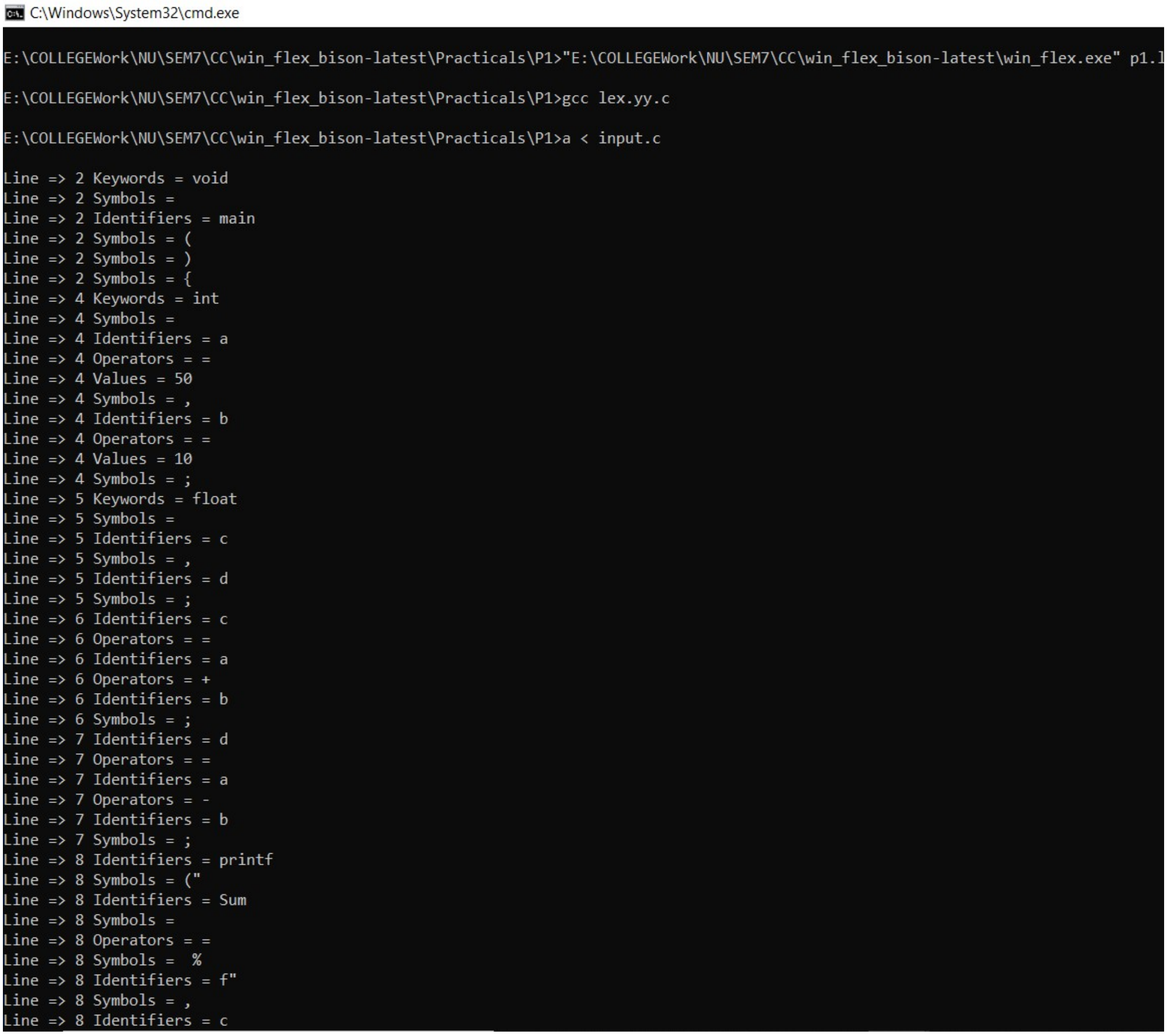
Input File:

```
void main(){

      int a=50,b=10;
      float c,d;
      c=a+b;
      d=a-b;
      printf("C = %f",c);
      printf("D = %f",d);

}
```

Output:



C:\Windows\System32\cmd.exe

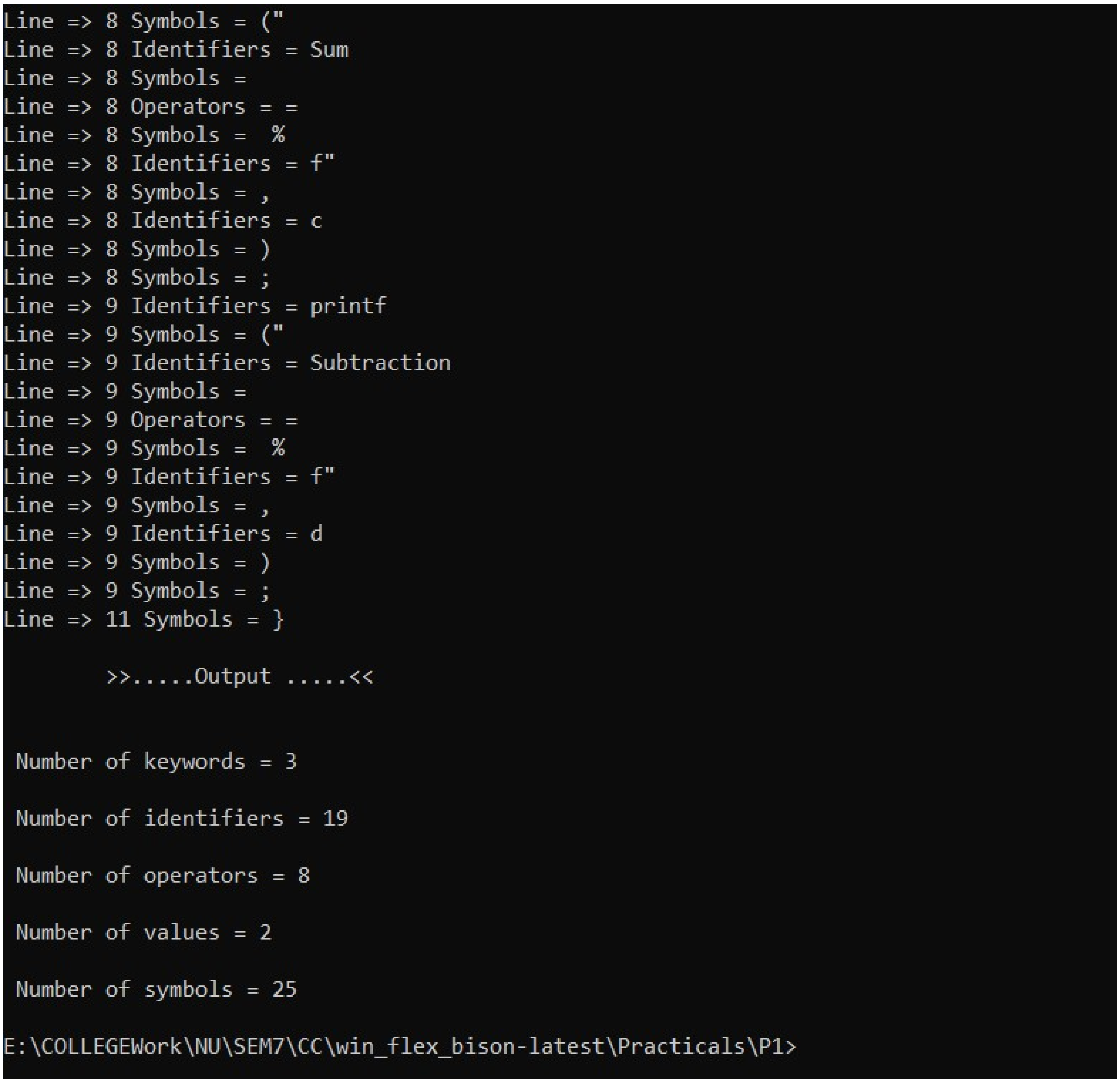
```
E:\COLLEGEWork\NU\SEM7\CC\win_flex_bison-latest\Practicals\P1>"E:\COLLEGEWork\NU\SEM7\CC\win_flex_bison-latest\win_flex.exe" p1.l

E:\COLLEGEWork\NU\SEM7\CC\win_flex_bison-latest\Practicals\P1>gcc lex.yy.c

E:\COLLEGEWork\NU\SEM7\CC\win_flex_bison-latest\Practicals\P1>a < input.c

Line => 2 Keywords = void
Line => 2 Symbols =
Line => 2 Identifiers = main
Line => 2 Symbols = (
Line => 2 Symbols = )
Line => 2 Symbols = {
Line => 4 Keywords = int
Line => 4 Symbols =
Line => 4 Identifiers = a
Line => 4 Operators = =
Line => 4 Values = 50
Line => 4 Symbols = ,
Line => 4 Identifiers = b
Line => 4 Operators = =
Line => 4 Values = 10
Line => 4 Symbols = ;
Line => 5 Keywords = float
Line => 5 Symbols =
Line => 5 Identifiers = c
Line => 5 Symbols = ,
Line => 5 Identifiers = d
Line => 5 Symbols = ;
Line => 6 Identifiers = c
Line => 6 Operators = =
Line => 6 Identifiers = a
Line => 6 Operators = +
Line => 6 Identifiers = b
Line => 6 Symbols = ;
Line => 7 Identifiers = d
Line => 7 Operators = =
Line => 7 Identifiers = a
Line => 7 Operators = -
Line => 7 Identifiers = b
Line => 7 Symbols = ;
Line => 8 Identifiers = printf
Line => 8 Symbols = ("
Line => 8 Identifiers = Sum
Line => 8 Symbols =
Line => 8 Operators = =
Line => 8 Symbols =  %
Line => 8 Identifiers = f"
Line => 8 Symbols = ,
Line => 8 Identifiers = c
```

Continued…



```
Line => 8 Symbols = ("
Line => 8 Identifiers = Sum
Line => 8 Symbols =
Line => 8 Operators = =
Line => 8 Symbols =  %
Line => 8 Identifiers = f"
Line => 8 Symbols = ,
Line => 8 Identifiers = c
Line => 8 Symbols = )
Line => 8 Symbols = ;
Line => 9 Identifiers = printf
Line => 9 Symbols = ("
Line => 9 Identifiers = Subtraction
Line => 9 Symbols =
Line => 9 Operators = =
Line => 9 Symbols =  %
Line => 9 Identifiers = f"
Line => 9 Symbols = ,
Line => 9 Identifiers = d
Line => 9 Symbols = )
Line => 9 Symbols = ;
Line => 11 Symbols = }

        >>.....Output .....<<


 Number of keywords = 3

 Number of identifiers = 19

 Number of operators = 8

 Number of values = 2

 Number of symbols = 25

E:\COLLEGEWork\NU\SEM7\CC\win_flex_bison-latest\Practicals\P1>
```

END