Name: Tushar Panchal

En.No: 21162101014

Sub: CD(Compiler Design)

Branch: CBA

Batch:71

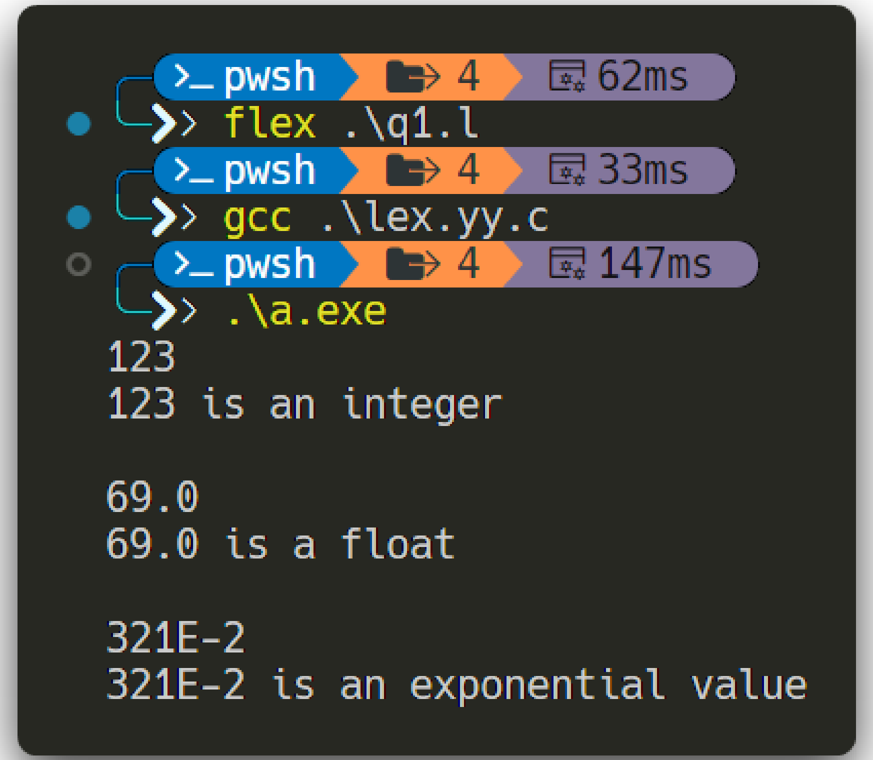
---------------------------PRACTICAL 04----------------------------

Write Lex Program
1) to Identify integer, Float and Exponential numbers
2)Identify Single and Multiline comments in C program
3)Identify valid tokens in given statement
scanf("%d %d",&a,&b);
printf("%d %d",a,b);

(1) <u>Source Code :</u>

```
%{
int INT = 0, FLOAT = 0, EP = 0;
%}
%%
[+-]?[0-9]+ { printf("%s is an integer\n", yytext); INT++; }
[+-]?[0-9]*[.][0-9]+ { printf("%s is a float\n", yytext); FLOAT++; }
[+-]?([0-9]+([.][0-9]*)?|[.][0-9]+)([eE][+-]?[0-9]+)? { printf("%s is
an exponential value\n", yytext); EP++; }
%%
int yywrap(){
    return 1;
}

int main() {
    yylex();
    printf("Number of\n1. Integer values: %d\n2. Float numbers: %d\n3.
Exponential values: %d\n", INT, FLOAT, EP);
    return 0;
}
```

✓ **Output :**

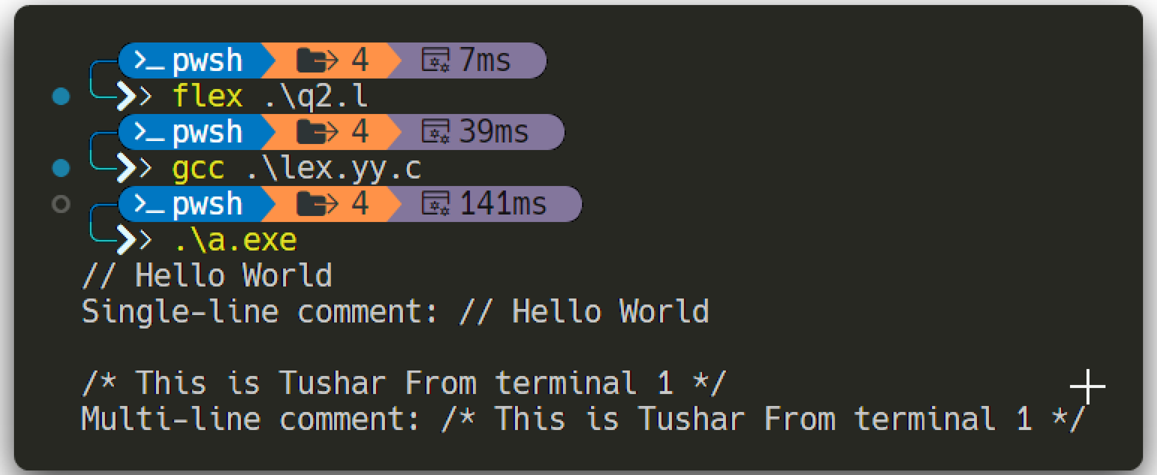


```
>_ pwsh  4  62ms
>> flex .\q1.l
>_ pwsh  4  33ms
>> gcc .\lex.yy.c
>_ pwsh  4  147ms
>> .\a.exe
123
123 is an integer

69.0
69.0 is a float

321E-2
321E-2 is an exponential value
```

(2) Source Code :

```
%{
#include <stdio.h>
%}
%%
"//".* {
    printf("Single-line comment: %s\n", yytext);
}
"/*"([^*]|\*+[^*/])*\*+"/" {
    printf("Multi-line comment: %s\n", yytext);
}
. {
    /* Ignore other characters */
}
%%
int yywrap() {
    return 1;
}

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

✓ **Output :**

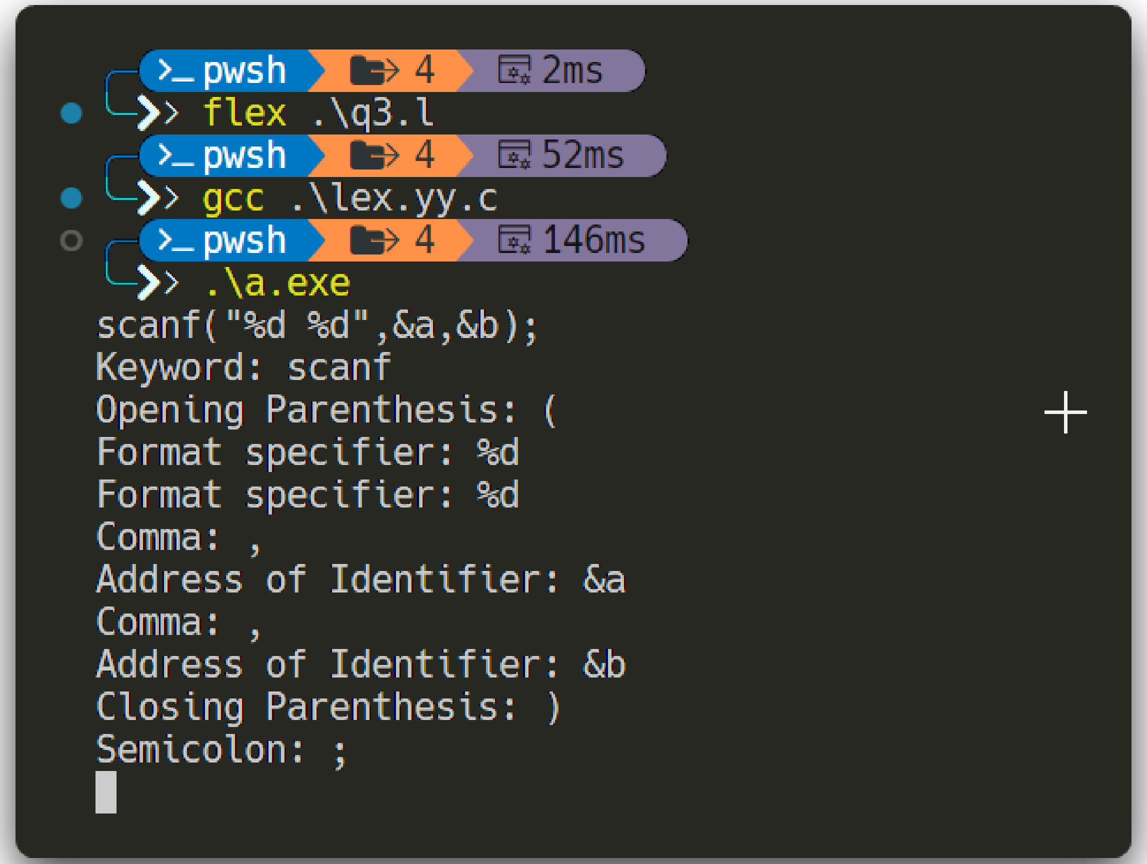


```
╭ >_ pwsh  4  7ms
╰>> flex .\q2.l
╭ >_ pwsh  4  39ms
╰>> gcc .\lex.yy.c
╭ >_ pwsh  4  141ms
╰>> .\a.exe
// Hello World
Single-line comment: // Hello World

/* This is Tushar From terminal 1 */
Multi-line comment: /* This is Tushar From terminal 1 */
```

(3) <u>Source Code :</u>

```
%{
#include <stdio.h>
%}
%%
"scanf" { printf("Keyword: %s\n", yytext); }
"printf" { printf("Keyword: %s\n", yytext); }
"%d" { printf("Format specifier: %s\n", yytext); }
"," { printf("Comma: %s\n", yytext); }
";" { printf("Semicolon: %s\n", yytext); }
"\"" { /* Ignore double quotes */ }
[ \t\n]+ { /* Ignore whitespace */ }
"&"[a-zA-Z_][a-zA-Z0-9_]* { printf("Address of Identifier: %s\n",
yytext); }
[a-zA-Z_][a-zA-Z0-9_]* { printf("Identifier: %s\n", yytext); }
"(" { printf("Opening Parenthesis: %s\n", yytext); }
")" { printf("Closing Parenthesis: %s\n", yytext); }
. { printf("Unrecognized text: %s\n", yytext); }
%%
int yywrap() {
    return 1;
}

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

✓ **Output :**



```
>_ pwsh  4  2ms
>> flex .\q3.l
>_ pwsh  4  52ms
>> gcc .\lex.yy.c
>_ pwsh  4  146ms
>> .\a.exe
scanf("%d %d",&a,&b);
Keyword: scanf
Opening Parenthesis: (
Format specifier: %d
Format specifier: %d
Comma: ,
Address of Identifier: &a
Comma: ,
Address of Identifier: &b
Closing Parenthesis: )
Semicolon: ;
```