



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Symbiosis International (Deemed University)

(Established under section 3 of the UGC Act, 1956)

Re-accredited by NAAC with 'A' grade (3.58/4) | Awarded Category – I by UGC

Founder: Prof. Dr. S. B. Mujumdar, M. Sc., Ph. D. (Awarded Padma Bhushan and Padma Shri by President of India)

Assignment No. 04

Subject:	Compiler Construction Lab
Name of Student	Onkar Mendhapurkar
PRN No.	22070122135
Branch	CSE B2, Batch (2022-26)
Academic Year & Semester	2022-26
Date of Performance	21/08/2025
Title of Assignment:	Conversion of lowercase to uppercase and vice versa.
Practice Questions	<ol style="list-style-type: none">LEX code for conversion of lowercase to uppercase and vice versa.LEX code to check whether the given character is in upper case, or in lower case or non-alphabetic character.LEX code to count the lowercase, upper case characters in the given input file.
Source Code	<pre>1. %{ #include <stdio.h> %} %% [a-z] { printf("%c", yytext[0] - 'a' + 'A'); } /* lowercase → uppercase */ [A-Z] { printf("%c", yytext[0] - 'A' + 'a'); } /* uppercase → lowercase */ . { printf("%s", yytext); } /* print everything else as-is */ \n { printf("\n"); } int yywrap() { return 1; }</pre>

```

2.
%{
#include <stdio.h>
%}

%%

[A-Z] { printf("%s' is an Uppercase character\n", yytext); }
[a-z] { printf("%s' is a Lowercase character\n", yytext); }
.    { printf("%s' is a Non-alphabetic character\n", yytext); }
\n   { /* Ignore newline */ }

%%

int yywrap() {
    return 1;
}

3.
%{
#include <stdio.h>
int upper_count = 0;
int lower_count = 0;
%}

%%

[A-Z] { upper_count++; }
[a-z] { lower_count++; }
.\n { /* ignore other characters */ }

%%

int yywrap() {
    return 1;
}

int main() {
    yylex();
    printf("Uppercase letters: %d\n", upper_count);
    printf("Lowercase letters: %d\n", lower_count);
    return 0;
}

```

Output Screenshot

1.

```

battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL$ mkdir Assign4
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL$ cd Assign4/
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ nano caseconvert.l
%{
#include <stdio.h>
%}

%%
[a-z]   { printf("%c", yytext[0] - 'a' + 'A'); } /* lowercase → uppercase */
[A-Z]   { printf("%c", yytext[0] - 'A' + 'a'); } /* uppercase → lowercase */
.     { printf("%s", yytext); }                   /* print everything else as-is */
\n     { printf("\n"); }

int yywrap() {
    return 1;
}
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ flex caseconvert.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ gcc lex.yy.c -o caseconvert -lfl
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ ./caseconvert
Hello World Onkar
hELLO wORLD oNKAR
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ |

```

2.

```

battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ nano charcheck.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ cat charcheck.l
%{
#include <stdio.h>
%}

%%
[A-Z]   { printf("'"%s' is an Uppercase character\n", yytext); }
[a-z]   { printf("'"%s' is a Lowercase character\n", yytext); }
.     { printf("'"%s' is a Non-alphabetic character\n", yytext); }
\n     { /* Ignore newline */ }

int yywrap() {
    return 1;
}
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ flex charcheck.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ gcc lex.yy.c -o charcheck -lfl
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ ./charcheck
A
'A' is an Uppercase character
z
'z' is a Lowercase character
5
'5' is a Non-alphabetic character
@
'@' is a Non-alphabetic character
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ |

```

3.

```

battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ nano countcase.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ flex countcase.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ gcc lex.yy.c -o countcase -lfl
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ echo "Hello World 123 ABC xyz" > input.txt
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ ./countcase < input.txt
Uppercase letters: 5
Lowercase letters: 11
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ cat countcase.l
%{
#include <stdio.h>
int upper_count = 0;
int lower_count = 0;
%}

%%
[A-Z]   { upper_count++; }
[a-z]   { lower_count++; }
.|\n   { /* ignore other characters */ }

int yywrap() {
    return 1;
}

int main() {
    yylex();
    printf("Uppercase letters: %d\n", upper_count);
    printf("Lowercase letters: %d\n", lower_count);
    return 0;
}

```

Post lab questions

1. LEX code for case conversion of alphabets/**alphanumeric** term using file handling.

Code:

```
%{
#include <stdio.h>
%}

%%%
[a-z] { printf("%c", yytext[0] - 'a' + 'A'); } /* lowercase → uppercase */
[A-Z] { printf("%c", yytext[0] - 'A' + 'a'); } /* uppercase → lowercase */
[0-9] { printf("%s", yytext); } /* digits remain same */
.\n { printf("%s", yytext); } /* other chars remain same */
%%%



int yywrap() {
    return 1;
}

int main(int argc, char *argv[]) {
    if (argc > 1) {
        FILE *f = fopen(argv[1], "r");
        if (!f) {
            perror("Error opening file");
            return 1;
        }
        yyin = f; /* redirect input stream */
    }
    yylex(); /* process input */
    return 0;
}
```

Output:

```

battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ nano filecaseconvert.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ cat filecaseconvert.l
%{
#include <stdio.h>
%}

<%
[a-z]   { printf("%c", yytext[0] - 'a' + 'A'); } /* lowercase → uppercase */
[A-Z]   { printf("%c", yytext[0] - 'A' + 'a'); } /* uppercase → lowercase */
[0-9]   { printf("%s", yytext); }                  /* digits remain same */
.\n    { printf("%s", yytext); }                  /* other chars remain same */

%%

int yywrap() {
    return 1;
}

int main(int argc, char *argv[]) {
    if (argc > 1) {
        FILE *f = fopen(argv[1], "r");
        if (!f) {
            perror("Error opening file");
            return 1;
        }
        yyin = f; /* redirect input stream */
    }
    yylex(); /* process input */
    return 0;
}
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ flex filecaseconvert.l
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ gcc lex.yy.c -o filecaseconvert -lfl
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ echo "Hello123 WorldXYZ test" > input.txt
battlemachine@DESKTOP-FU1975B:/mnt/c/Users/DELL/CCL/Assign4$ ./filecaseconvert input.txt
Hello123 wORLDxyz TEST

```

Conclusion	These LEX programs show how powerful lexical analysis can be for text processing. From converting cases and classifying characters to counting and handling files, LEX efficiently scans and transforms input using simple rules. Overall, it demonstrates how pattern matching and automation make LEX useful beyond compiler design in real-world applications.
------------	---