Compiler Construction Lab Practicals Document

Submitted by

Poorva Pohekar, 22070521108

B. Tech Computer Science and Engineering

Under the Guidance of

Dr.Snehlata K Wankhade

Symbiosis Institute of Technology, Nagpur.

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS

INSTITUTE OF TECHNOLOGY, NAGPUR

Wathoda, Nagpur
2025

PRACTICAL -01
Aim: Theory Assignment for writing details about Lex and Yacc compiler

Code:

```
%{
#include<stdio.h>
%}
%%
zero|ZERO printf("0");
one|ONE printf("1");
two|TWO printf("2");
three|THREE printf("3");
four|FOUR printf("4");
five|FIVE printf("5");
six|SIX printf("6");
seven|SEVEN printf("7");
eight|EIGHT printf("8");
nine|NINE printf("9");
%%
int main(){
yylex();
return 0;
};
```

Gedit Code(number_1108.l):

Open | number_1108.l ~/Desktop/compiler_construction | Save

```
%{
#include<stdio.h>
%}
%%
zero|ZERO printf("0");
one|ONE printf("1");
two|TWO printf("2");
three|THREE printf("3");
four|FOUR printf("4");
five|FIVE printf("5");
six|SIX printf("6");
seven|SEVEN printf("7");
eight|EIGHT printf("8");
nine|NINE printf("9");
%%
int main(){
yylex();
return 0;
};
```

Output:

```
(base) [avroop@arch compiler_construction]$ lex number_1108.l
(base) [avroop@arch compiler_construction]$ cc lex.yy.c -lfl
(base) [avroop@arch compiler_construction]$ ./a.out
zero
0
one
1
two
2
three
3
four
4
five
5
SIX
6
SEVEN
7
EIGTH
EIGTH
EIGHT
8
nine
9
```

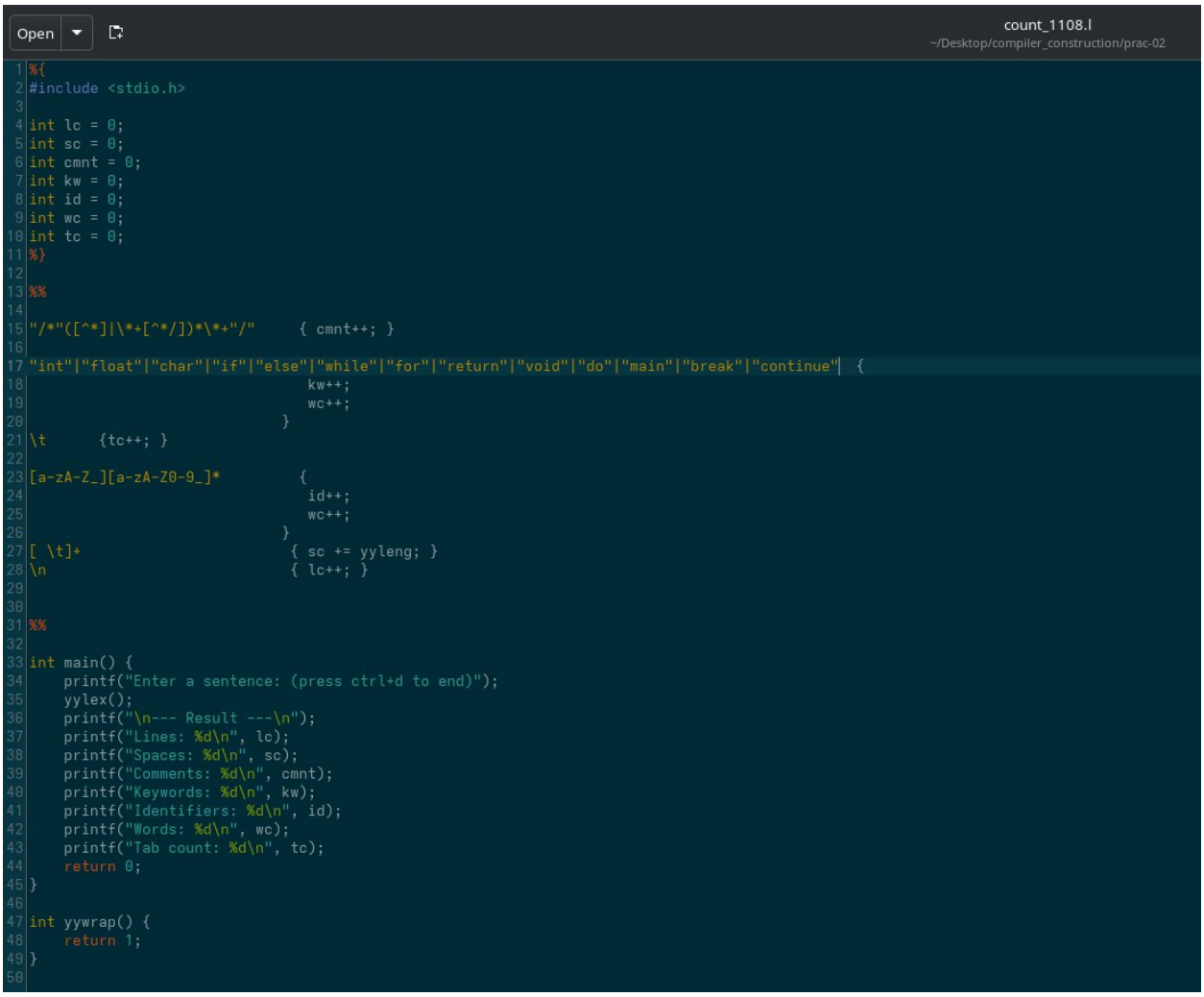
Practical-02
Aim: Count the numbers of comments, keywords, identifiers, word and space line from users input.

Code:

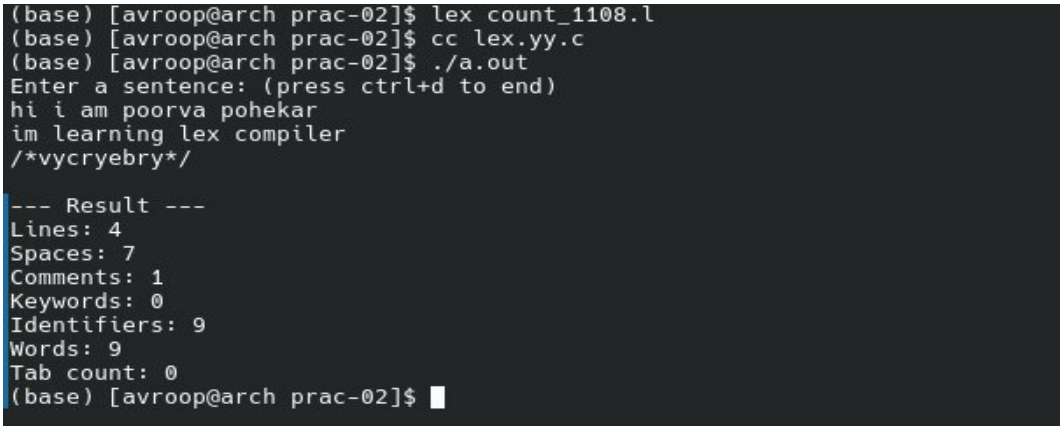
```
%{
#include <stdio.h>
int lc = 0;
int sc = 0;
int cmnt = 0;
int kw = 0;
int id = 0;
int wc = 0;
int tc = 0;
%}

%%
"/*"([^*]|\*+[^*/])*\*+"/"     { cmnt++; }
"int"|"float"|"char"|"if"|"else"|"while"|"for"|"return"|"void"|"do"|"main"|"break"|"continue"  {
                                kw++;
                                wc++;
                              }
\t          {tc++; }
[a-zA-Z_][a-zA-Z0-9_]*         {
                                id++;
                                wc++;
                              }
[ \t]+                         { sc += yyleng; }
\n                             { lc++; }
%%
int main() {
    printf("Enter a sentence: (press ctrl+d to end)");
    yylex();
    printf("\n--- Result ---\n");
    printf("Lines: %d\n", lc);
    printf("Spaces: %d\n", sc);
    printf("Comments: %d\n", cmnt);
    printf("Keywords: %d\n", kw);
    printf("Identifiers: %d\n", id);
    printf("Words: %d\n", wc);
    printf("Tab count: %d\n", tc);
    return 0;
```

```
}
int yywrap() {
    return 1;
}
```

Gedit Code(count_1108.l):



Output:



Prcatical-03

Aim: To write lex program that counts number of words starting with the letter ‘a’ or ‘A’ from given input text.

Code:

```
%{
#include <stdio.h>
int count = 0;
%}
%%
[aA][a-zA-Z]*     { count++; }          // Count word starting with 'a' or 'A'
[^a-zA-Z]+        { /* Skip non-words */ }
[a-zA-Z]+         { /* Skip other words */ }
%%
int yywrap() {
    return 1;
}
int main() {
    printf("Enter text (Press Ctrl+D to end input):\n");
    yylex();
    printf("\nNumber of words starting with 'a' or 'A': %d\n", count);
    return 0;
}
```

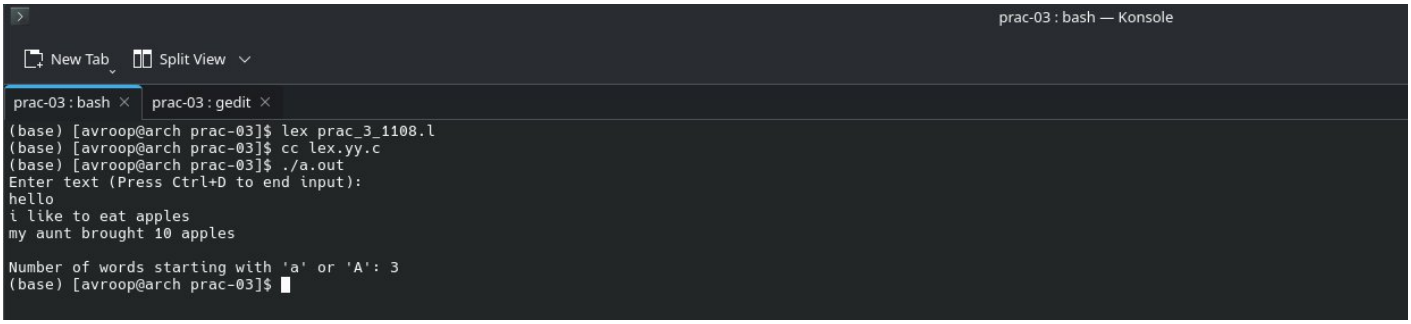
Gedit Code(prac_3_1108.l):

Open

prac_3_1108.l
~/Desktop/compiler_construction/prac-03

```
%{
#include <stdio.h>
int count = 0;
%}

%%

[aA][a-zA-Z]*     { count++; }          // Count word starting with 'a' or 'A'
[^a-zA-Z]+        { /* Skip non-words */ }
[a-zA-Z]+         { /* Skip other words */ }

%%

int yywrap() {
    return 1;
}

int main() {
    printf("Enter text (Press Ctrl+D to end input):\n");
    yylex();
    printf("\nNumber of words starting with 'a' or 'A': %d\n", count);
    return 0;
}
```

Output:



Practical-04

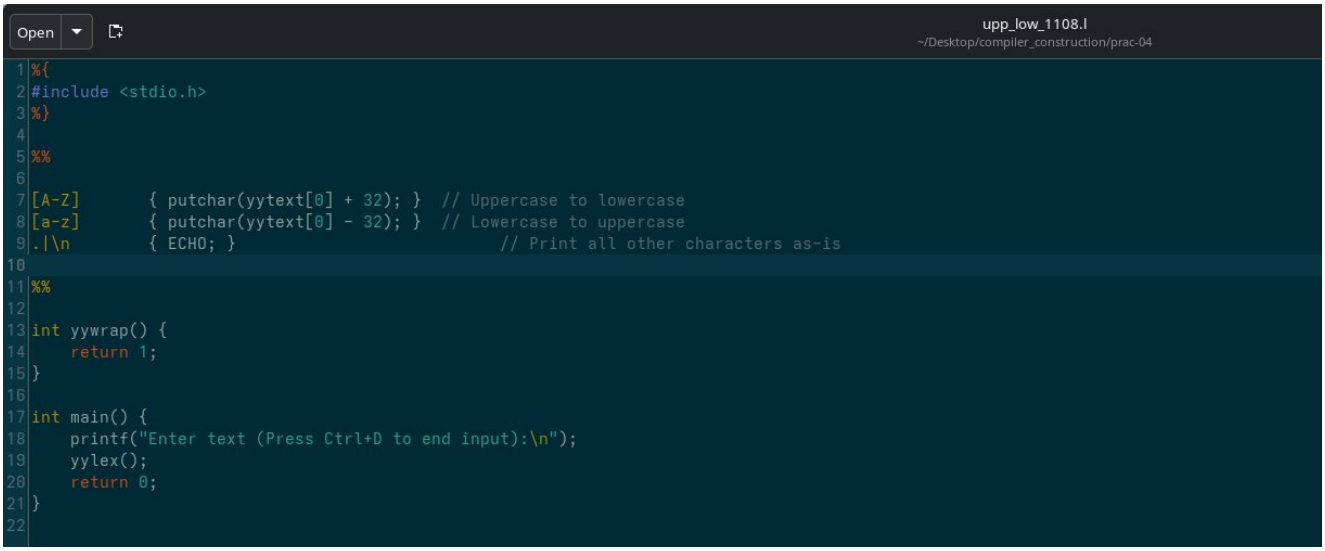
Aim: Conversion of lowercase to uppercase and vice-versa.

Code:

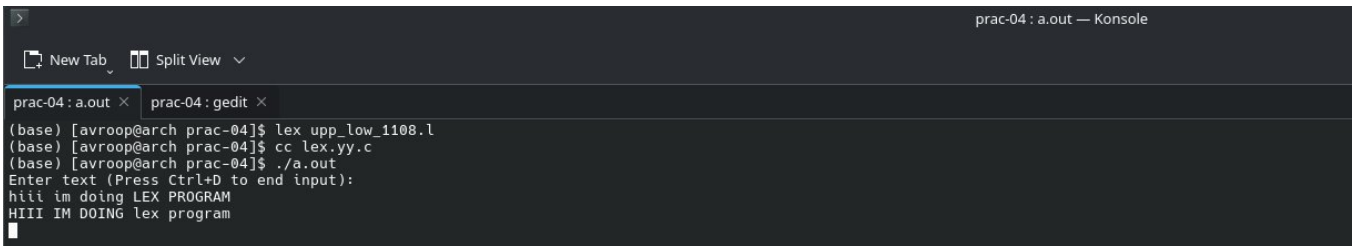
```
%{
#include <stdio.h>
%}

%%
[A-Z]      { putchar(yytext[0] + 32); }  // Uppercase to lowercase
[a-z]      { putchar(yytext[0] - 32); }  // Lowercase to uppercase
```

```
.|\n        { ECHO; }                       // Print all other characters as-is
%%
int yywrap() {
    return 1;
}
int main() {
    printf("Enter text (Press Ctrl+D to end input):\n");
    yylex();
    return 0;
}
```

Gedit Code(upp_low_1108.l):



Output:



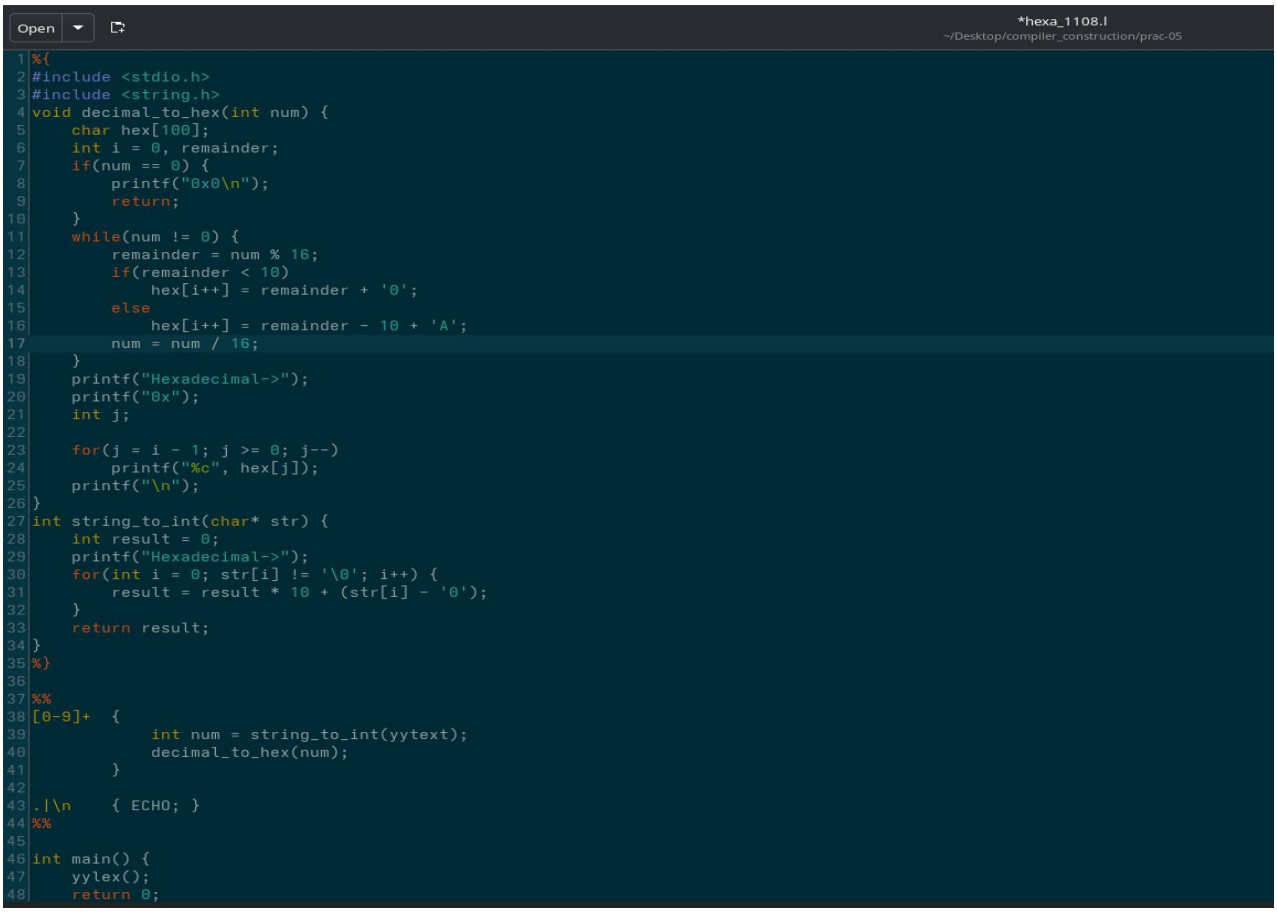
Practical-05
Aim: Conversion of Decimal number to it’s Hexadecimal equivalent from user input.

Code:

```
%{
#include <stdio.h>
#include <string.h>
void decimal_to_hex(int num) {
    char hex[100];
    int i = 0, remainder;
    if(num == 0) {
        printf("0x0\n");
        return;
    }
    while(num != 0) {
        remainder = num % 16;
        if(remainder < 10)
            hex[i++] = remainder + '0';
        else
            hex[i++] = remainder - 10 + 'A';
        num = num / 16;
    }
    printf("0x");
    int j;
    for(j = i - 1; j >= 0; j--)
        printf("%c", hex[j]);
    printf("\n");
}
int string_to_int(char* str) {
    int result = 0;
    for(int i = 0; str[i] != '\0'; i++) {
        result = result * 10 + (str[i] - '0');
    }
    return result;
}
%}

%%
```

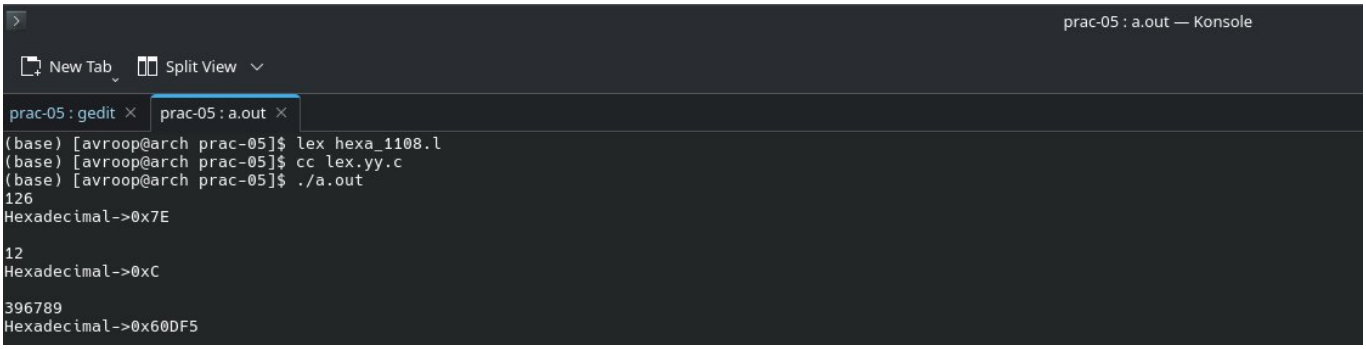
```
[0-9]+  {
            int num = string_to_int(yytext);
            decimal_to_hex(num);
        }

.|\n    { ECHO; }
%%

int main() {
    yylex();
    return 0;
}
int yywrap() {
    return 1;
}
```

Gedit Code(hexa_1108.l):



```
%{
#include <stdio.h>
#include <string.h>
void decimal_to_hex(int num) {
    char hex[100];
    int i = 0, remainder;
    if(num == 0) {
        printf("0x0\n");
        return;
    }
    while(num != 0) {
        remainder = num % 16;
        if(remainder < 10)
            hex[i++] = remainder + '0';
        else
            hex[i++] = remainder - 10 + 'A';
        num = num / 16;
    }
    printf("Hexadecimal->");
    printf("0x");
    int j;

    for(j = i - 1; j >= 0; j--)
        printf("%c", hex[j]);
    printf("\n");
}
int string_to_int(char* str) {
    int result = 0;
    printf("Hexadecimal->");
    for(int i = 0; str[i] != '\0'; i++) {
        result = result * 10 + (str[i] - '0');
    }
    return result;
}
%}

%%
[0-9]+  {
            int num = string_to_int(yytext);
            decimal_to_hex(num);
        }

.|\n    { ECHO; }
%%

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

Output:



Practical-06

Aim: To write a Lex program that identifies and counts lines ending with the substring "com" from the input text and prints the matched lines.

Code:

```
%{
#include <stdio.h>
int ln=0;
%}
%%
.*com$   { ln++; printf("Matched line: %d %s",ln, yytext); }
.*
%%
int main() {
    printf("Enter text here: \n");
    yylex();
    return 0;
}
int yywrap() {
    return 1;
}
```

Gedit Code(com_1108.l):

Open

com_1108.l
~/Desktop/compiler_construction/prac-06

```
%{
#include <stdio.h>
int ln=0;
%}

%%
.*com$   { ln++; printf("Matched line: %d %s",ln, yytext); }
.*
%%

int main() {
    printf("Enter text here: \n");
    yylex();
    return 0;
}
int yywrap() {
    return 1;
}
```

Output:

prac-06 : a.out — Konsole

New Tab Split View

New Tab (Ctrl+Shift+T)

prac-06 : a.out prac-06 : gedit

```
(base) [avroop@arch prac-06]$ lex com_1108.l
(base) [avroop@arch prac-06]$ cc lex.yy.c
(base) [avroop@arch prac-06]$ ./a.out
Enter text here:
hi

i am poorva

my id is pogvtwvy@gmail.com
Matched line: 1 my id is pogvtwvy@gmail.com
```