



## **Assignment**

Course Title: **Compiler Design Lab**

Course Code: CSE332

### **Submitted by**

**MD. Moshiur Rahman**

ID: 0242220005101871

Section: 63\_M2

Semester: Fall -25

Department: Computer Science and Engineering

### **Submitted to**

**Md. Rashedul Alam**

Lecturer

Computer Science & Engineering

Daffodil International University

**Submission Date: 14 December 2025**

## Table of Contents

Sl. No.	Experiment Title
1	<b>Program to count the length of a string</b>
2	<b>Program to count the number of white spaces in a given string</b>
3	<b>Program to remove white spaces from a string</b>
4	<b>Program to count vowels, consonants and digits in a string</b>
5	<b>Program to tokenize a string using strtok()</b>
6	<b>Program to tokenize a string without using strtok()</b>
7	<b>Program to take multiple lines as input and count the number of lines</b>
8	<b>Program to take multiple lines as input and identify the comments</b>
9	<b>Program to take multiple lines as input and remove single-line/multi-line comments</b>
10	<b>Program to identify articles from a given input string and count the articles</b>
11	<b>Program to recognize whether the input is a valid identifier or not</b>
12	<b>Program to find and count the maximum frequency of a word in a given string</b>
13	<b>Program to find the maximum frequency of a word in a given string</b>
14	<b>Program to count the length of a string and reverse with an extra message</b>
15	<b>Program to count characters without white space</b>
16	<b>Program to enter 3 characters and show the next 3 characters</b>
17	<b>Program to show the initial of a name (e.g., “Mushfiqur Rahman” → “MR”)</b>
18	<b>Program to show syntax table (letters, digits, symbols, arithmetic/logical operators)</b>
19	<b>Program to remove special characters from a string</b>
20	<b>Program to test regular expressions with examples (various patterns over {a, b})</b>

## 1. Length of a String

```
#include <stdio.h>
int main(){
    char str[200];
    int i,len=0;
    fgets(str,200,stdin);
    for(i=0;str[i]!='\0' && str[i]!='\n';i++)
        len++;
    printf("Length = %d",len);
    return 0;
}
```

### Output:

```
Moshiur
Length = 7

...Program finished with exit code 0
Press ENTER to exit console. █
```

## 2. Count White Spaces

```
#include <stdio.h>
#include <ctype.h>
int main(){
    char str[200];
    int i,space=0;
    fgets(str,200,stdin);
    for(i=0;str[i]!='\0';i++)
        if(isspace(str[i])) space++;
    printf("Spaces = %d",space);
    return 0;
}
```

### Output:

```
Moshiur Rahman
Spaces = 1

...Program finished with exit code 0
Press ENTER to exit console. █
```

## 3. Remove White Spaces

```
#include <stdio.h>
#include <ctype.h>
int main(){
    char str[200],res[200];
    int i,j=0;
    fgets(str,200,stdin);
```

```
for(i=0;str[i]!='\0';i++)
    if(!isspace(str[i]))
        res[j++]=str[i];
res[j]='\0';
printf("%s",res);
return 0;
}
```

**Output:**

```
This is Moshiur
ThisisMoshiur

...Program finished with exit code 0
Press ENTER to exit console.
```

#### 4. Count Vowels, Consonants & Digits

```
#include <stdio.h>
#include <ctype.h>
int main(){
    char str[200];
    int v=0,c=0,d=0,i;
    fgets(str,200,stdin);
    for(i=0;str[i];i++){
        if(isalpha(str[i])){
            char ch=tolower(str[i]);
            if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u') v++;
            else c++;
        }else if(isdigit(str[i])) d++;
    }
    printf("Vowels=%d Consonants=%d Digits=%d",v,c,d);
    return 0;
}
```

**Output:**

```
Moshiur
Vowels=3 Consonants=4 Digits=0

...Program finished with exit code 0
Press ENTER to exit console.
```

## 5. Tokenization using strtok

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[200];
    char *token;
    const char *delim = " ,;:\t\n";
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Tokens:\n");
    token = strtok(str, delim);
    while (token != NULL) {
        printf("%s\n", token);
        token = strtok(NULL, delim);
    }
    return 0;
}
```

### Output:

```
Enter a string: Moshiur Rahman
Moshiur
Rahman
```

```
==== Code Execution Successful ===
```

## 6. Tokenization without strtok

```
#include <stdio.h>

int is_delim(char c) {
    char delims[] = " ,;:\t\n";
    for (int i = 0; delims[i]; i++)
        if (c == delims[i]) return 1;
    return 0;
}

int main() {
    char str[200];
    int i = 0, start = -1;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
```

```

printf("Tokens:\n");
while (str[i] != '\0') {
    if (!is_delim(str[i]) && start == -1)
        start = i;

    if ((is_delim(str[i]) || str[i+1]=='\0') && start != -1) {
        int end = is_delim(str[i]) ? i : i+1;
        for (int j = start; j < end; j++)
            putchar(str[j]);
        putchar('\n');
        start = -1;
    }
    i++;
}
return 0;
}

```

**Output:**

```

Enter a string: MD MOSHIUR RAHMAN
MD
MOSHIUR
RAHMAN

```

```

==== Code Execution Successful ====

```

## 7. Count Number of Lines

```

#include <stdio.h>
#include <string.h>

int main() {
    char line[200];
    int count = 0;

    printf("Enter multiple lines (type END to stop):\n");
    while (1) {
        fgets(line, sizeof(line), stdin);
        line[strcspn(line, "\n")] = '\0';

        if (strcmp(line, "END") == 0)
            break;

        count++;
    }

    printf("Number of lines: %d\n", count);
    return 0;
}

```

**Output:**

```
Enter multiple lines (type END to stop):
Moshiur
shemul
Shihab
END
Number of lines: 3

==== Code Execution Successful ====
```

## 8. Identify Comments

```
#include <stdio.h>
#include <string.h>

int main() {
    char line[300];
    int in_block = 0;

    printf("Enter C code (type END to stop):\n");

    while (1) {
        fgets(line, sizeof(line), stdin);
        if (strncmp(line, "END", 3) == 0)
            break;

        for (int i = 0; line[i]; i++) {
            if (!in_block && line[i]=='/' && line[i+1]== '/') {
                printf("%s", &line[i]);
                break;
            }
            if (!in_block && line[i]=='/' && line[i+1]=='*') {
                in_block = 1;
                printf("/*");
                i++;
            }
            else if (in_block) {
                putchar(line[i]);
                if (line[i]=='*' && line[i+1]== '/') {
                    putchar('/');
                    in_block = 0;
                    i++;
                }
            }
        }
    }
    return 0;
}
```

**Output:**

```
Enter C code (type END to stop):
int a; // variable
END

// variable

==== Code Execution Successful ====
```

## 9. Remove Comments

```
#include <stdio.h>

int main() {
    char line[300];
    int in_block = 0;

    printf("Enter C code (type END to stop):\n");

    while (1) {
        fgets(line, sizeof(line), stdin);
        if (line[0]=='E' && line[1]=='N' && line[2]=='D')
            break;

        for (int i = 0; line[i]; i++) {
            if (in_block) {
                if (line[i]=='*' && line[i+1]== '/') {
                    in_block = 0;
                    i++;
                }
                continue;
            }

            if (line[i]=='/' && line[i+1]=='*') {
                in_block = 1;
                i++;
                continue;
            }

            if (line[i]=='/' && line[i+1]=='/')
                break;

            putchar(line[i]);
        }
    }
    return 0;
}
```

**Output:**

```

Enter C code (type END to stop):
int a; // comment
END

int a;

==== Code Execution Successful ====

```

## 10. Identify Articles

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char str[300], word[50];
    int i = 0, j = 0, count = 0;

    printf("Enter a sentence: ");
    fgets(str, sizeof(str), stdin);

    while (str[i] != '\0') {
        if (isalpha((unsigned char)str[i])) {
            word[j++] = tolower(str[i]);
        } else {
            if (j > 0) {
                word[j] = '\0';
                if (strcmp(word, "a") == 0 ||
                    strcmp(word, "an") == 0 ||
                    strcmp(word, "the") == 0) {
                    printf("Article found: %s\n", word);
                    count++;
                }
                j = 0;
            }
        }
        i++;
    }

    printf("Total articles: %d\n", count);
    return 0;
}

```

**Output:**

```

Enter a sentence: The cat is on a table

Article found: the
Article found: a
Total articles: 2

==== Code Execution Successful ====

```

## 11. Valid Identifier

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>

int is_valid_identifier(const char *s) {
    int i;

    if (!(isalpha((unsigned char)s[0]) || s[0] == '_'))
        return 0;

    for (i = 1; s[i] != '\0'; i++) {
        if (!(isalnum((unsigned char)s[i]) || s[i] == '_'))
            return 0;
    }
    return 1;
}

int main() {
    char id[100];

    printf("Enter an identifier: ");
    scanf("%99s", id);

    if (is_valid_identifier(id))
        printf("\"%s\" is a valid identifier.\n", id);
    else
        printf("\"%s\" is NOT a valid identifier.\n", id);

    return 0;
}
```

### Output:

```
Enter an identifier: total_sum

"total_sum" is a valid identifier.

==== Code Execution Successful ====
```

## 12. Max Word Frequency

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_WORDS 100
#define MAX_LEN 50

int main() {
    char str[500];
```

```

char words[MAX_WORDS][MAX_LEN];
int count[MAX_WORDS] = {0};
int word_index = 0;
int i = 0;

printf("Enter a sentence: ");
fgets(str, sizeof(str), stdin);

while (str[i] != '\0') {
    char temp[MAX_LEN];
    int k = 0;

    while (isalpha((unsigned char)str[i])) {
        temp[k++] = tolower(str[i]);
        i++;
    }
    temp[k] = '\0';

    if (k > 0) {
        int found = -1;
        for (int j = 0; j < word_index; j++) {
            if (strcmp(words[j], temp) == 0) {
                found = j;
                break;
            }
        }
        if (found == -1) {
            strcpy(words[word_index], temp);
            count[word_index]++;
        } else {
            count[found]++;
        }
    } else {
        i++;
    }
}

int max = 0, pos = -1;
for (i = 0; i < word_index; i++) {
    if (count[i] > max) {
        max = count[i];
        pos = i;
    }
}

printf("Word with maximum frequency: %s\n", words[pos]);
printf("Frequency: %d\n", max);

return 0;
}

```

**Output:**

```
Enter a sentence: C is fun and C is powerful
```

```
Word with maximum frequency: c
Frequency: 2
```

```
==== Code Execution Successful ====
```

### 13. Maximum Frequency Only

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_WORDS 200
#define MAX_LEN 50

int main() {
    char str[500];
    char words[MAX_WORDS][MAX_LEN];
    int count[MAX_WORDS] = {0};
    int total_words = 0;

    printf("Enter a sentence: ");
    fgets(str, sizeof(str), stdin);

    int i = 0;
    while (str[i] != '\0') {
        char temp[MAX_LEN];
        int j = 0;

        while (isalpha((unsigned char)str[i])) {
            temp[j++] = tolower(str[i]);
            i++;
        }
        temp[j] = '\0';

        if (j > 0) {
            int found = -1;
            for (int k = 0; k < total_words; k++) {
                if (strcmp(words[k], temp) == 0) {
                    found = k;
                    break;
                }
            }

            if (found == -1) {
                strcpy(words[total_words], temp);
                count[total_words++] = 1;
            } else {
                count[found]++;
            }
        }
    }
}
```

```

        }
        i++;
    }

int max = 0;
for (int a = 0; a < total_words; a++) {
    if (count[a] > max)
        max = count[a];
}
printf("Maximum frequency of any word: %d\n", max);
return 0;
}

```

**Output:**

```
Enter a sentence: one two two three three three
```

```
Maximum frequency of any word: 3
```

```
==== Code Execution Successful ===
```

## 14. Reverse String

```
#include <stdio.h>

int main() {
    char str[200], rev[200];
    int len = 0, i, j;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    while (str[len] != '\0' && str[len] != '\n')
        len++;

    for (i = len - 1, j = 0; i >= 0; i--, j++)
        rev[j] = str[i];

    rev[j] = '\0';

    printf("Original length of the string: %d\n", len);
    printf("Reversed string: %s\n", rev);
    return 0;
}
```

**Output:**

```
Enter a string: Compiler

Original length of the string: 8
Reversed string: relipmoC

==== Code Execution Successful ====
```

## 15. Count Characters Without Space

```
#include <stdio.h>
#include <ctype.h>

int main() {
    char str[200];
    int i, count = 0;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    for (i = 0; str[i] != '\0'; i++) {
        if (!isspace((unsigned char)str[i]))
            count++;
    }

    printf("Number of characters (excluding white spaces): %d\n", count);
    return 0;
}
```

### Output:

```
Enter a string: C Programming

Number of characters (excluding white spaces): 12

==== Code Execution Successful ====
```

## 16. Next Characters

```
#include <stdio.h>
int main() {
    char c1, c2, c3
    printf("Enter 3 characters: ");
    scanf(" %c %c %c", &c1, &c2, &c3)
    printf("Next 3 characters: %c %c %c\n",
           c1 + 1, c2 + 1, c3 + 1);
    return 0;
}
```

**Output:**

```
Enter 3 characters: a b c

Next 3 characters: b c d

==== Code Execution Successful ====
```

## 17. Print Initials

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main() {
    char name[200];
    int i;
    printf("Enter full name: ");
    fgets(name, sizeof(name), stdin);
    printf("Initials: ")
    if (isalpha((unsigned char)name[0]))
        printf("%c", toupper(name[0]));
    for (i = 1; name[i] != '\0'; i++) {
        if (name[i - 1] == ' ' && isalpha((unsigned char)name[i])) {
            printf("%c", toupper(name[i]));
        }
    }
    printf("\n");
    return 0;
}
```

**Output:**

```
Enter full name: md moshiur rahman

Initials: MMR

==== Code Execution Successful ====
```

## 18. Classify Expression

```
#include <stdio.h>
#include <ctype.h>
int is_arith(char c) {
    return (c=='+' || c=='-' || c=='*' || c=='/' || c=='%');
}
```

```

int is_logic(char c) {
    return (c=='&' || c=='|' || c=='!' || c=='^');
}
int main() {
    char str[300];
    int i;
    printf("Enter an expression: ");
    fgets(str, sizeof(str), stdin);
    printf("Letters: ");
    for (i = 0; str[i] != '\0'; i++)
        if (isalpha((unsigned char)str[i]))
            printf("%c ", str[i]);
    printf("\nDigits: ");
    for (i = 0; str[i] != '\0'; i++)
        if (isdigit((unsigned char)str[i]))
            printf("%c ", str[i]);
    printf("\nArithmetic operators: ");
    for (i = 0; str[i] != '\0'; i++)
        if (is_arith(str[i]))
            printf("%c ", str[i]);

    printf("\nLogical operators: ");
    for (i = 0; str[i] != '\0'; i++)
        if (is_logic(str[i]))
            printf("%c ", str[i]);
    printf("\nSymbols: ");
    for (i = 0; str[i] != '\0'; i++)
        if (!isalnum((unsigned char)str[i]) &&
            !isspace((unsigned char)str[i]) &&
            !is_arith(str[i]) &&
            !is_logic(str[i]))
            printf("%c ", str[i]);
    printf("\n");
    return 0;
}

```

**Output:**

Enter an expression: a+b>=c1&d

Letters: a b c d  
 Digits: 1  
 Arithmetic operators: +  
 Logical operators: &  
 Symbols: > =

==== Code Execution Successful ===

## 19. Remove Special Characters

```
#include <stdio.h>
#include <ctype.h>
int main() {
    char str[300], res[300];
    int i, j = 0;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    for (i = 0; str[i] != '\0'; i++) {
        if (isalnum((unsigned char)str[i]) ||
            isspace((unsigned char)str[i])) {
            res[j++] = str[i];
        }
    }
    res[j] = '\0';
    printf("String without special characters: %s\n", res);
    return 0;
}
```

### Output:

```
Enter a string: C@Prog#2025!
```

```
String without special characters: CProg2025
```

```
==== Code Execution Successful ====
```

## 20. Regular Expression Checker

```
#include <stdio.h>
#include <string.h>
/* (ab)* */
int match_ab_star(const char *s) {
    int i = 0;
    if (s[0] == '\0') return 1;
    while (s[i] != '\0') {
        if (s[i] != 'a' || s[i+1] != 'b')
            return 0;
        i += 2;
    }
    return 1;
}
/* ab* */
int match_ab_star_single(const char *s) {
    if (s[0] != 'a') return 0;
    for (int i = 1; s[i] != '\0'; i++)
        if (s[i] != 'b') return 0;
    return 1;
}
/* ab+ */
int match_ab_plus(const char *s) {
```

```

if (s[0] != 'a' || s[1] != 'b') return 0;
for (int i = 2; s[i] != '\0'; i++)
    if (s[i] != 'b') return 0;
    return 1;
}

/* a(a|b)*a */
int match_a_aorb_star_a(const char *s) {
    int len = strlen(s);
    if (len < 2) return 0;
    if (s[0] != 'a' || s[len-1] != 'a') return 0;
    for (int i = 1; i < len-1; i++)
        if (s[i] != 'a' && s[i] != 'b') return 0;
    return 1;
}
/* (a|b)*b(a|b)*b(a|b) */
int match_two_bs(const char *s) {
    int len = strlen(s), bcount = 0;
    if (len < 3) return 0;
    for (int i = 0; i < len; i++) {
        if (s[i] != 'a' && s[i] != 'b') return 0;
        if (s[i] == 'b') bcount++;
    }
    return (bcount >= 2);
}
/* (a|b)*a(a|b)(a|b)* */
int match_a_in_middle(const char *s) {
    int len = strlen(s);
    if (len < 2) return 0;
    for (int i = 0; i < len; i++)
        if (s[i] != 'a' && s[i] != 'b') return 0;
    for (int i = 0; i < len-1; i++)
        if (s[i] == 'a') return 1;
    return 0;
}
int main() {
    char str[100];
    int choice, result = 0
    printf("Regular Expression Tester over {a,b}\n");
    printf("1. (ab)*\n");
    printf("2. ab*\n");
    printf("3. ab+\n");
    printf("4. a(a|b)*a\n");
    printf("5. (a|b)*b(a|b)*b(a|b)\n");
    printf("6. (a|b)*a(a|b)(a|b)*\n");
    printf("Enter your choice (1-6): ");
    scanf("%d", &choice);
    printf("Enter a string of a's and b's: ");
    scanf("%s", str);
    switch (choice) {
        case 1: result = match_ab_star(str); break;
        case 2: result = match_ab_star_single(str); break;
        case 3: result = match_ab_plus(str); break;
        case 4: result = match_a_aorb_star_a(str); break;
        case 5: result = match_two_bs(str); break;
        case 6: result = match_a_in_middle(str); break;
        default:
    }
}

```

```
    printf("Invalid choice\n");
    return 0;
}
if (result)
    printf("String is ACCEPTED by the expression.\n");
else
    printf("String is NOT accepted by the expression.\n");
return 0;
}
```

**Output:**

```
Regular Expression Tester over {a,b}
1. (ab)*
2. ab*
3. ab+
4. a(a|b)*a
5. (a|b)*b(a|b)*b(a|b)
6. (a|b)*a(a|b)(a|b)*
Enter your choice (1-6): 4
Enter a string of a's and b's: abba

String is ACCEPTED by the expression.
```

```
==== Code Execution Successful ===
```