

**OPEN SOURCE SOFTWARE LAB (15B17CI575)**  
**Lab Assignment 1 (Practice Lab)**  
**Odd 2025**

Profile link: <https://github.com/PranshuSharma14>

2. Create a C program that reads an array of integers from the user, calculates the average of the integers, and prints the result. Initialize a new Git repository for this project, commit the code, and upload it to your GitHub account. Provide the URL of the repository.

```
#include <stdio.h>

int main() {
    int n, i;
    float sum = 0;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        sum += arr[i];
    }

    float average = sum / n;
    printf("Average = %.2f\n", average);

    return 0;
}
```

```
C:\Users\pavni\OneDrive\Des > + | ^

Enter the number of elements: 3
Enter 3 integers:
3
4
5
Average = 4.00

Process returned 0 (0x0) execution time : 10.429 s
Press any key to continue.
|
```

3. Write a C program that takes a string input from the user and reverses the string. Initialize a Git repository, commit your changes, and push the repository to GitHub. Share the repository link.

```
C:\Users\pavni\OneDrive\Des > + | ^

Enter a string: helloworld
Reversed string: dlrowolleh

Process returned 0 (0x0) execution time : 6.900 s
Press any key to continue.
|
```

4. Extract one open source project from Github. Perform the reverse engineering of the same.

Project: Snake Game in C

Reverse Engineering Goal: Understand the structure, game loop, input handling, snake movement, and rendering logic.

Outcome: You've broken down a working game into understandable components and can now recreate, modify, or improve it.

5. Find an open-source C project on GitHub that implements a basic calculator. Download the project, analyze its code structure, and document how each function works. Initialize a new Git repository, commit your documentation, and upload it to your GitHub account. Provide the URL of the repository.

Breif: This repository contains a simple C program that takes a string input from the user and prints the reversed string. It demonstrates basic string handling and loop concepts in C, making it a good starting point for beginners learning about arrays and input/output operations.

**OPEN SOURCE SOFTWARE LAB (15B17CI575)**  
**Lab Assignment 1b (Practice Lab)**  
**Odd 2025**

2. Create a C program that reads a matrix of integers from the user, calculates the transpose of the matrix, and prints the result. Initialize a new Git repository for this project, commit the code, and upload it to your GitHub account. Provide the URL of the repository.

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

```
int main() {
    int rows, cols;

    printf("Enter the number of rows and columns of the matrix: ");
    scanf("%d %d", &rows, &cols);

    int matrix[rows][cols];
    int transpose[cols][rows];

    printf("Enter elements of the matrix:\n");
    for(int i = 0; i < rows; i++) {
        for(int j = 0; j < cols; j++) {
            printf("Element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }

    for(int i = 0; i < rows; i++) {
        for(int j = 0; j < cols; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }

    printf("\nTranspose of the matrix:\n");
    for(int i = 0; i < cols; i++) {
        for(int j = 0; j < rows; j++) {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

```

Enter the number of rows and columns of the matrix: 3
3
Enter elements of the matrix:
Element [1][1]: 1
Element [1][2]: 2
Element [1][3]: 3
Element [2][1]: 4
Element [2][2]: 5
Element [2][3]: 6
Element [3][1]: 7
Element [3][2]: 8
Element [3][3]: 9

Transpose of the matrix:
1 4 7
2 5 8
3 6 9

Process returned 0 (0x0)    execution time : 32.117 s
Press any key to continue.

```

3. Write a C program that takes a paragraph input from the user, counts the frequency of each word in the paragraph, and prints the frequencies. Initialize a Git repository, commit your changes, and push the repository to GitHub. Share the repository link.

```

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

#define MAX_WORDS 100
#define MAX_WORD_LEN 50
#define MAX_PARAGRAPH_LEN 1000

typedef struct {
    char word[MAX_WORD_LEN];
    int count;
} WordFreq;

int main() {
    char paragraph[MAX_PARAGRAPH_LEN];
    WordFreq words[MAX_WORDS];
    int wordCount = 0;

    printf("Enter a paragraph:\n");
    fgets(paragraph, sizeof(paragraph), stdin);

```

```

for (int i = 0; paragraph[i]; i++) {
    paragraph[i] = tolower(paragraph[i]);
}

const char *delimiters = " ,.!?\n\t;";
char *token = strtok(paragraph, delimiters);

while (token != NULL) {
    int found = 0;

    for (int i = 0; i < wordCount; i++) {
        if (strcmp(words[i].word, token) == 0) {
            words[i].count++;
            found = 1;
            break;
        }
    }

    if (!found && wordCount < MAX_WORDS) {
        strcpy(words[wordCount].word, token);
        words[wordCount].count = 1;
        wordCount++;
    }
}

token = strtok(NULL, delimiters);
}

printf("\nWord Frequencies:\n");
for (int i = 0; i < wordCount; i++) {
    printf("%s: %d\n", words[i].word, words[i].count);
}

return 0;
}

```

```
C:\Users\pavni\OneDrive\Desktop + ▾
Enter a paragraph:
hi my name is pavni. i am in btech 3rd year

Word Frequencies:
hi: 1
my: 1
name: 1
is: 1
pavni: 1
i: 1
am: 1
in: 1
btech: 1
3rd: 1
year: 1

Process returned 0 (0x0)  execution time : 353.139 s
Press any key to continue.
|
```

4. Find an open-source C project on GitHub that implements a graph algorithm (e.g., Dijkstra's algorithm). Download the project, analyze its code structure, and document how each function works. Initialize a new Git repository, commit your documentation, and upload it to your GitHub account. Provide the URL of the repository.