Assignment 1

Student Id – 2403S0839
BSc (Hons) in Computer Science – Batch 03

1.

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

int main() {
    int MAX_STUDENTS = 100;
    int MAX_NAME_LENGTH = 60;

    FILE *fp;
    char names[100][60];
    float grades[100];
    char line[100];
    int count = 0;
    float sum = 0, average;

    fp = fopen("math_grades.txt", "r"); // Open the file

    if (fp == NULL) {
        printf("Error: Unable to open the file math_grades.txt. Please check if the file exists.\n");
        return 1;
    } // Check if file opened successfully

    while (fgets(line, sizeof(line), fp)) {
        char name[60];
        float grade; // Read the file line by line

        sscanf(line, "%s %f", name, &grade); // Read the name and grade (we can still use scanf style
here)

        int j = 0;
        while (name[j] != '\0') {
            names[count][j] = name[j];
            j++;
        } // Manual string copy (character by character)

        names[count][j] = '\0'; // End of string

        grades[count] = grade;

        sum = sum + grade;
        count++;
    }

    fclose(fp);

    average = sum / count; // Calculate average

    printf("Average grade: %.2f\n\n", average); // Display average
    printf("Students who scored above average:\n");
    for (int i = 0; i < count; i++) {
        if (grades[i] > average) {
            printf("%s\n", names[i]);
        }
    } // Students above average

    printf("\nStudents who scored below or equal to average:\n");
    for (int i = 0; i < count; i++) {
        if (grades[i] <= average) {
            printf("%s\n", names[i]);
        }
    } // Students below or equal to average

    return 0;
}
```

2. i. The program opens the file and checks if it is opened correctly.
ii. Then it reads the file one line at a time.
iii. Each line has a student’s name and grade.
iv. The program takes the name and grade from the line and saves them in two separate lists: one for names and one for grades.
v. It repeats this until all students are read and close the file.

3. i. The program adds all the grades together in a variable called sum.
ii. After adding all the grades, it divides the sum by the total number of students.
iii. This gives the average grade.

4. i. Loops help the program repeat actions many times, like reading all students or checking each student’s grade.
ii. If statements help the program make decisions, for example, to check if the file is open or if a grade is above the average.
iii. These structures help the program run correctly and handle different situations.

5. i. The average grade will be about 82.36.
John, Bob, and Mary students have above grade. And Alice and David have grades below or equal to 82.36.
ii. The program will print the average and list these students in two groups.

6. i. If the file cannot be opened, the program will show a message like: “Error: Unable to open the file math_grades.txt. Please check if the file exists.”
ii. Then the program stops to avoid further errors.

7. i. Let the program read other subjects’ files.
ii. Sort students by name or grade for easier reading.
iii. Save the results in a new file to keep records.
iv. Check if grades are valid numbers between 0 and 100.