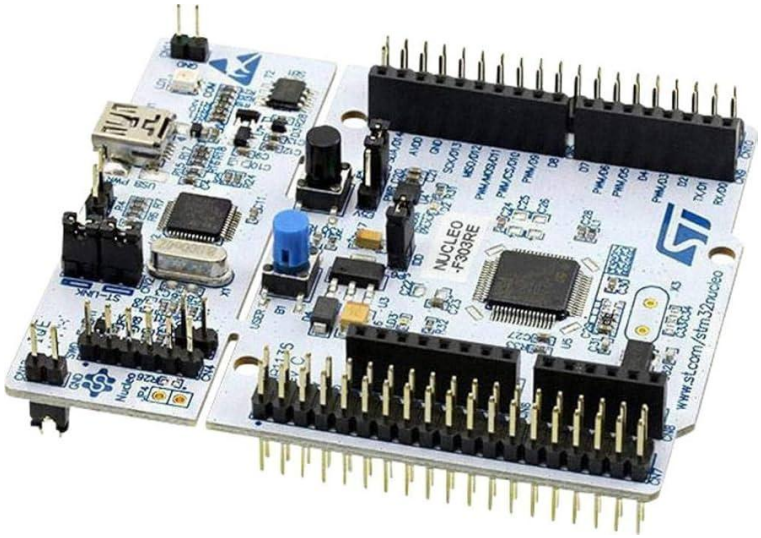
CPE 2200 Hardware Programming

Lab 3: Data Analysis in C



By: Marlon Morales
Kennesaw State University
July 10, 2025

KENNESAW STATE
UNIVERSITY

CPE 2200 Laboratory Exercise Cover Sheet

Name (print): Marlon Morales

Date: July 10, 2025

Lab Title: Data Analysis in C

Lab Number: Lab 3

Who You Helped: N/A

Who Helped You: N/A

Lab Checklist:

☑ Did you fill out the cover sheet?

☑ Is there a flow diagram of your algorithm on the next page?

☑ Is your single .c code commented?

☑ Will you upload your single .c file?

☑ Will you submit this cover sheet and the flow-diagram as a single 1-pdf to D2L along with your separate .c file?

___For Remote ONLY (*must receive prior approval*): Will you upload to D2L a video of you demonstrating your lab working for each test vector?

Demoed after Class

Flow Diagram

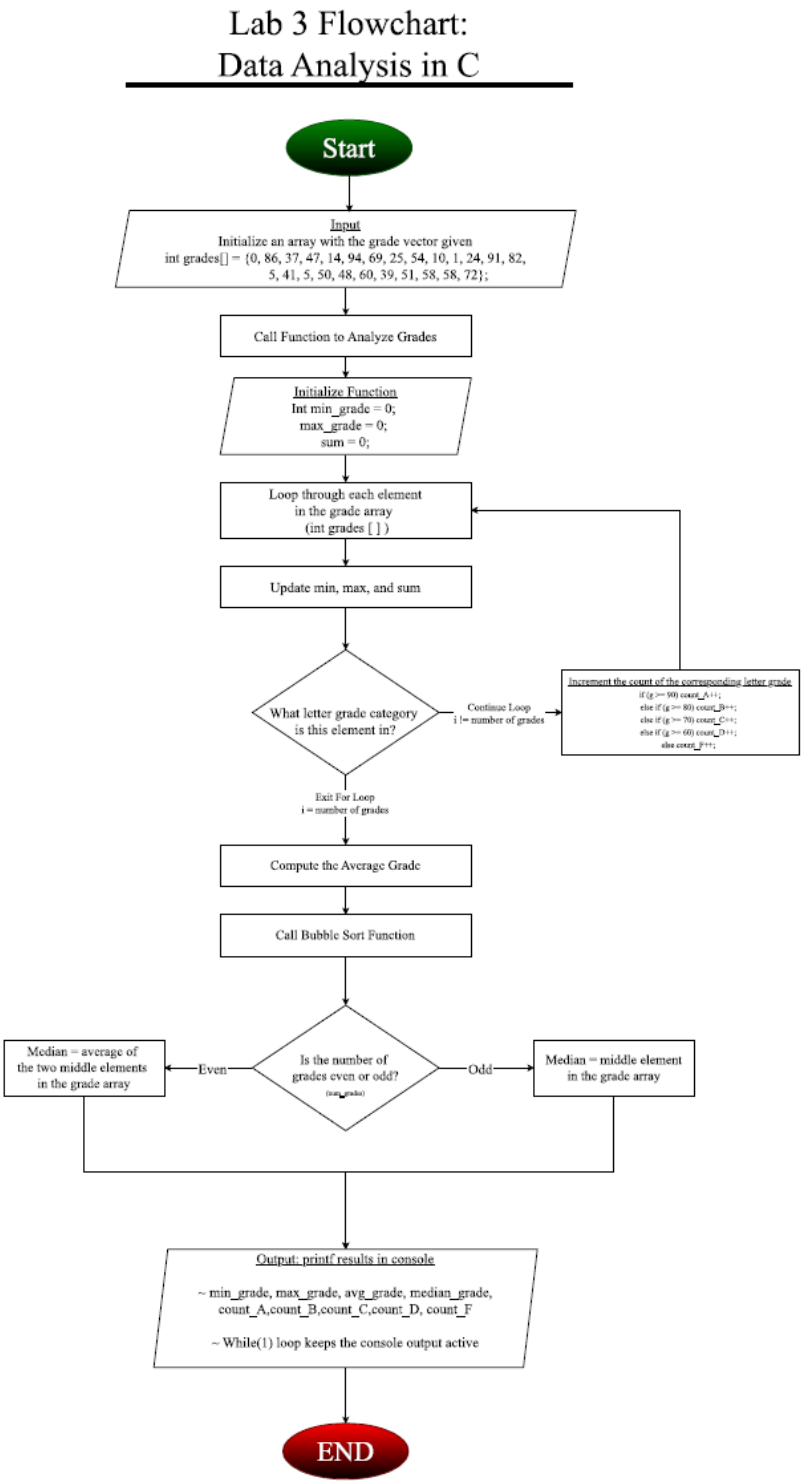


Figure 1:Flowchart created in Draw IO

C Code

```c
/*
 * Name: Marlon Morales
 * Date: [07/10/2025]
 * Course: CPE 2200
 * Lab 3 - Data Analysis in C Programming Language
 * Description: This program analyzes an array of grades, computes the minimum,
 * maximum, average, and median, and counts the number of A, B, C, D, and F letter
 * grades in the array.
 */

#include <stdio.h>
#include <stddef.h>  // for size_t type

// Semihosting setup (for printf output to IDE console)
extern void initialise_monitor_handles(void);

// --------- Functions ---------
// Declare functions before main so they can be defined after main
void swap(int *a, int *b);
void bubble_sort(int arr[], size_t num_elements);
void analyze_grades(int grades[], size_t num_grades);

// --------- Global Variables ---------
// Hold the computed results (accessible by all functions)
int min_grade, max_grade, avg_grade, median_grade;
int count_A = 0, count_B = 0, count_C = 0, count_D = 0, count_F = 0;

int main(void) {
    // Initialize semihosting (needed for printf to work over debugger)
    initialise_monitor_handles();

    // --------- Input Grades ---------
    int grades[] = {0, 86, 37, 47, 14, 94, 69, 25, 54, 10, 1, 24, 91, 82,
                    5, 41, 5, 50, 48, 60, 39, 51, 58, 58, 72};
    const size_t NUM_GRADES = sizeof(grades) / sizeof(grades[0]);  // count of grades

    // --------- Analyze Grades ---------
    analyze_grades(grades, NUM_GRADES);

    // --------- Print Results ---------
    printf("\nGrade Analysis Results\n");
    printf("----------------------\n");
    printf("Min Grade        : %d\n", min_grade);
    printf("Max Grade        : %d\n", max_grade);
    printf("Average Grade    : %d\n", avg_grade);
    printf("Median Grade     : %d\n", median_grade);
    printf("Count A (90-100): %d\n", count_A);
    printf("Count B (80-89) : %d\n", count_B);
    printf("Count C (70-79) : %d\n", count_C);
    printf("Count D (60-69) : %d\n", count_D);
    printf("Count F (<60)   : %d\n", count_F);

    while (1) {      // Loop

    }
}

// --------- Swap Function ---------
// Swaps two integers in memory (used in bubble sort)
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```c
// --------- Bubble Sort Function ---------
// Simple bubble sort to sort grades array in ascending order
void bubble_sort(int arr[], size_t num_elements) {
    for (size_t k = 0; k < num_elements - 1; k++) {
        for (size_t i = 0; i < num_elements - k - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                swap(&arr[i], &arr[i + 1]);
            }
        }
    }
}

// --------- Grade Analysis Function ---------
// Computes min, max, average, median, and counts letter grade categories
void analyze_grades(int grades[], size_t num_grades) {
    min_grade = grades[0];
    max_grade = grades[0];
    int sum = 0;

    // Loop through grades to calculate min, max, sum, and grade counts
    for (size_t i = 0; i < num_grades; i++) {
        int g = grades[i];
        if (g < min_grade) min_grade = g;
        if (g > max_grade) max_grade = g;
        sum += g;

        // Count letter grades by range
        if (g >= 90) count_A++;
        else if (g >= 80) count_B++;
        else if (g >= 70) count_C++;
        else if (g >= 60) count_D++;
        else count_F++;
    }

    // Calculate average (integer division, rounds down)
    avg_grade = sum / num_grades;

    // Sort the grades to prepare for median calculation
    bubble_sort(grades, num_grades);

    // Calculate median
    if (num_grades % 2 == 1) {
        // Odd number of grades → middle element
        median_grade = grades[num_grades / 2];
    } else {
        // Even number of grades → average of two middle elements
        int mid1 = grades[(num_grades / 2) - 1];
        int mid2 = grades[num_grades / 2];
        median_grade = (mid1 + mid2) / 2;
    }
}
```

Results Displayed to Console

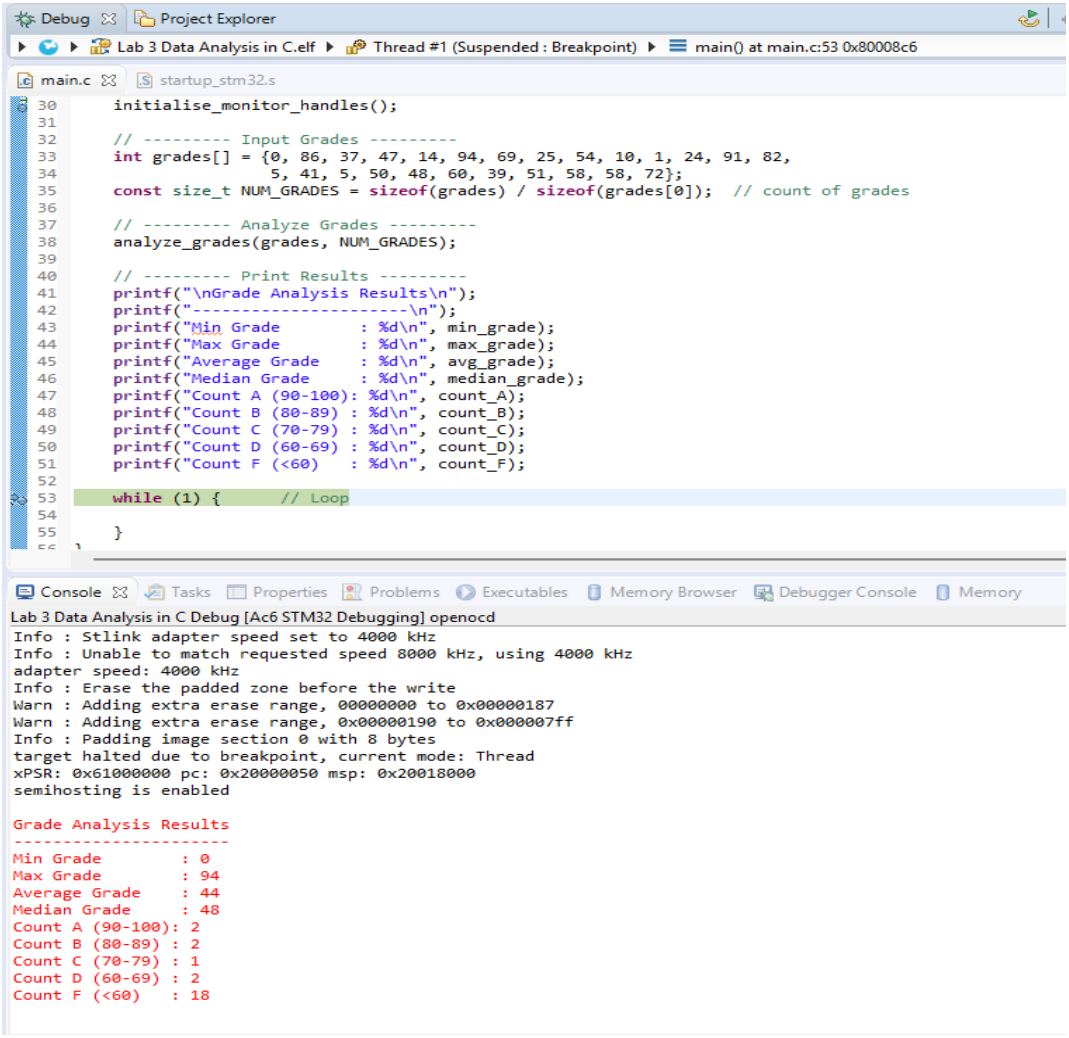


Figure 2:eclipse IDE & console

```
Grade Analysis Results
----------------------
Min Grade         : 0
Max Grade         : 94
Average Grade     : 44
Median Grade      : 48
Count A (90-100): 2
Count B (80-89)  : 2
Count C (70-79)  : 1
Count D (60-69)  : 2
Count F (<60)    : 18
```

Figure 3: eclipse IDE Console