



## Digital Design Verification

### FYP/Internship Program

LAB # 01

**GCC Compiler, GNU Debugger and Data Types,**

**Name: Abdul Hadi**

**Afzal**

**Date: 25-02-2025**

## Table of Contents

Task 1: .....	3
PART 1 (Default): .....	3
PART 1 (Errors Fixed): .....	3
PART 2 : .....	3
PART 3: .....	4
PART 4: (Introducing an error): .....	4
PART 4 (Fixing the error): .....	5
Task 2: .....	7
Task 3: .....	8
Task 4: .....	9
Task 5: .....	9
Task 6: .....	10
Task 7: .....	10
Conclusion:.....	12

## Task 1:

### PART 1 (Default):

```

Tab@NCDC-0019 MSYS /c/Users/lab/Desktop/LAB1
$ cd /c/Users/lab/Desktop/LAB1
gcc pwd_checker.c test_pwd_checker.c -o pwd_checker
./pwd_checker
pwd_checker.c: In function 'check_upper':
pwd_checker.c:36:21: warning: comparison between pointer and zero character constant [-Wpointer-comp
are]
 36 |     while (password != '\0') {
      |     ^
pwd_checker.c:36:12: note: did you mean to dereference the pointer?
 36 |     while (password != '\0') {
      |     ^
Running tests...

Assertion failed: test1, file test_pwd_checker.c, line 12

Tab@NCDC-0019 MSYS /c/Users/lab/Desktop/LAB1
$ cd /c/Users/lab/Desktop/LAB1
gcc pwd_checker.c test_pwd_checker.c -o pwd_checker
./pwd_checker
pwd_checker.c: In function 'check_upper':
pwd_checker.c:36:21: warning: comparison between pointer and zero character constant [-Wpointer-comp
are]
 36 |     while (password != '\0') {
      |     ^
pwd_checker.c:36:12: note: did you mean to dereference the pointer?
 36 |     while (password != '\0') {
      |     ^
pwd_checker.c: In function 'check_lower':
pwd_checker.c:48:21: warning: comparison between pointer and zero character constant [-Wpointer-comp
are]
 48 |     while (password != '\0') {
      |     ^
pwd_checker.c:48:12: note: did you mean to dereference the pointer?
 48 |     while (password != '\0') {
      |     ^
pwd_checker.c: In function 'check_number':
pwd_checker.c:60:21: warning: comparison between pointer and zero character constant [-Wpointer-comp
are]
 60 |     while (password != '\0') {
      |     ^
pwd_checker.c:60:12: note: did you mean to dereference the pointer?
 60 |     while (password != '\0') {
      |     ^
pwd_checker.c: At top level:
pwd_checker.c:78:3: error: expected ';' before '_Bool'
 78 | }z
      |
      |
Running tests...

Assertion failed: test1, file test_pwd_checker.c, line 12

```

### PART 1 (Errors Fixed):

```

    ,,_----_
Running tests...

Congrats! The first test case passed.

Assertion failed: meets_len_req, file pwd_checker.c, line 11

```

### PART 2 :

```
$ gcc -Wall -Wextra -g pwd_checker.c test_pwd_checker.c -o pwd_checker
pwd_checker.c: In function 'check_password':
pwd_checker.c:62:5: error: implicit declaration of function 'assert' [-Wimplicit-function-declaration]
  62 |     assert(lower); // Debugging check for lowercase
     |     ^
pwd_checker.c:6:1: note: 'assert' is defined in header '<assert.h>'; this is probably fixable by adding '#include <assert.h>'
    5 | #include "pwd_checker.h"
    +++ |+#include <assert.h>
     6 |
```

```
Running tests...
Congrats! The first test case passed.
Assertion failed: meets_len_req, file pwd_checker.c, line 11
```

## PART 3:

```
$ cd /c/Users/lab/Desktop/LAB1
gcc -Wall -Wextra -g pwd_checker.c test_pwd_checker.c -o pwd_checker
./pwd_checker
Running tests...

Assertion failed: test1, file test_pwd_checker.c, line 13
```

```
Breakpoint 2 at 0x1400014d2: file pwd_checker.c, line 21.
(gdb) run
Starting program: C:\Users\lab\Desktop\LAB1\pwd_checker.exe
[New Thread 1648.0x560]
Running tests...

Thread 1 hit Breakpoint 2, check_upper (password=0x7ff651744021 "qrtv?,mp!ltrA0b13rab4ham")
  at pwd_checker.c:21
21      while (*password != '\0') {
(gdb)
```

```
Thread 1 hit Breakpoint 2, check_upper (password=0x7ff651744021 "qrtv?,mp!ltrA0b13rab4ham")
  at pwd_checker.c:21
21      while (*password != '\0') {
(gdb) print password
$1 = 0x7ff651744021 "qrtv?,mp!ltrA0b13rab4ham"
(gdb)
$2 = 0x7ff651744021 "qrtv?,mp!ltrA0b13rab4ham"
(gdb)
```

```
$ ./pwd_checker
Running tests...
Congrats! The first test case passed.
Congrats! You have passed all of the test cases!
```

## PART 4: (Introducing an error):

```
/* Returns true if PASSWORD contains at least one number */
bool check_number(const char *password) {
    while (*password != '\0') {
        if (check_range(*password, '0', '9')) {
            return true;
        }
        ++password;
    }
    return false;
}
```

```
$1 = 9 '\t'
(gdb) print lower
$2 = 0 '\000'
(gdb)
```

## PART 4 (Fixing the error):

```
/* Returns true if PASSWORD contains at least one number */
bool check_number(const char *password) {
    while (*password != '\0') {
        if (check_range(*password, '0', '9')) {
            return true;
        }
        ++password;
    }
    return false;
}
```

```
gdb) print lower
1 = 48 '0'
gdb) print upper
2 = 57 '9'
(gdb)
```

```
Running tests...
Congrats! The first test case passed.
Congrats! You have passed all of the test cases!
```

## Final Code:

### Pwd\_checker.c:

```
*****
* Author: AbdulHadi Afzal
* Task: Password Strength Checker
*****
```

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include "pwd_checker.h"
```

```

bool check_length(const char *password) {
int length = strlen(password);
return (length >= 10);
}

bool check_range(char letter, char lower, char upper) {
return (letter >= lower && letter <= upper);
}

bool check_upper(const char *password) {
while (*password != '\0') {
if (check_range(*password, 'A', 'Z')) {
return true;
}
password++;
}
return false;
}

bool check_lower(const char *password) {
while (*password != '\0') {
if (check_range(*password, 'a', 'z')) {
return true;
}
++password;
}
return false;
}

bool check_number(const char *password) {
while (*password != '\0') {
if (check_range(*password, '0', '9')) {
return true;
}
++password;
}
return false;
}

bool check_name(const char *first_name, const char *last_name,
const char *password) {
return !(strstr(password, first_name) || strstr(password,
last_name));
}

/* Ensures password meets all security conditions */
bool check_password(const char *first_name, const char
*last_name, const char *password) {
return check_length(password) && check_upper(password) &&
check_lower(password) && check_number(password) &&
check_name(first_name, last_name, password);}

```

**test\_pwd\_checker.c:**

```
*****
 * Author: AbdulHadi Afzal          *
 * Task: Password Checker Test Cases   *
*****/
```

```
#include <assert.h>
#include <stdio.h>
#include "pwd_checker.h"

int main() {
    printf("Running tests...\n\n");

    // Test Case 1: This password should pass all checks
    const char *test1_first = "Abraham";
    const char *test1_last = "Garcia";
    const char *test1_pwd = "qrtv?,mp!ltrA0b13rab4ham";
    bool test1 = check_password(test1_first, test1_last, test1_pwd);
    assert(test1);
    printf("First test case passed.\n\n");

    // Test Case 2: Too short (Fails check_length)
    const char *test2_first = "John";
    const char *test2_last = "Doe";
    const char *test2_pwd = "Short1";
    bool test2 = check_password(test2_first, test2_last, test2_pwd);
    assert(!test2);

    // Test Case 3: No uppercase letter (Fails check_upper)
    const char *test3_pwd = "alllowercasel";
    bool test3 = check_password(test2_first, test2_last, test3_pwd);
    assert(!test3);

    // Test Case 4: No number (Fails check_number)
    const char *test4_pwd = "NoNumbersHere";
    bool test4 = check_password(test2_first, test2_last, test4_pwd);
    assert(!test4);

    // Test Case 5: Contains first name (Fails check_name)
    const char *test5_pwd = "JohnRocks123";
    bool test5 = check_password(test2_first, test2_last, test5_pwd);
    assert(!test5);

    printf("All test cases passed successfully.\n");
    return 0;
}
```

## Task 2:

```
$ gcc -o open_task2 open_task2.c
open_task2.c:1:10: fatal error: iostream: No such file or directory
  1 | #include <iostream>
    |
compilation terminated.
```

```
*****
 * Author: AbdulHadi Afzal          *
 * Task: Character Operations & Data Sizes   *
*****/
```

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

int main() {
    char chChar;

    printf("Enter a character: ");
    scanf(" %c", &chChar);

    printf("You entered: %c\n", chChar);
    printf("ASCII Value: %d\n", chChar);

    chChar = chChar + 7;
    printf("After adding 7: %c\n", chChar);

    chChar = 'd';
    printf("Final character: %c\n", chChar);

    printf("\n----- Data Type Sizes -----");
    printf("Size of char: %lu byte\n", sizeof(char));
    printf("Size of int: %lu bytes\n", sizeof(int));
    printf("Size of float: %lu bytes\n", sizeof(float));
    printf("Size of double: %lu bytes\n", sizeof(double));

    return 0;
}
```

```
$ gcc -o open_task2 open_task2.c
./open_task2
Enter a character: A
You entered: A
ASCII Value: 65
After adding 7: H
Final character: d

----- Data Type Sizes -----
Size of char: 1 byte
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
```

### Task 3:

```
*****
* Author: AbdulHadi Afzal
* Task: ASCII Name & CMS ID Display
*****
```

```
#include <stdio.h>

int main() {
    char lastCharName = 'i'; // Last letter of "Hadi"
    char lastCharCMS = '0'; // Last digit of CMS ID "413970"

    // Display Name using ASCII values
    printf("Name: %c%c%c%c\n", 72, 97, 100, lastCharName);
```

```

    // Display CMS ID
    printf("CMS ID: %d%d%d%d%c\n", 4, 1, 3, 9, 7, lastCharCMS);

    return 0;
}
  
```

```

$ gcc -o three three.c
./three
Name: Hadi
CMS ID: 413970
  
```

## Task 4:

```

/******************
 * Author: AbdulHadi Afzal
 * Task: Integer Arithmetic Operations
******************/

#include <stdio.h>

int main() {
    // Performing integer arithmetic calculations
    printf("35 / 5 = %d\n", 35 / 5);
    printf("36 / 7 = %d\n", 36 / 7);
    printf("18 - 32 / 6 * 3 = %d\n", 18 - 32 / 6 * 3);
    printf("220 / 5 = %d\n", 220 / 5);
    printf("27 - 7%3 + 8/3 = %d\n", 27 - 7 % 3 + 8 / 3);

    return 0;
}
  
```

```

$ gcc -o four four.c
./four
35 / 5 = 7
36 / 7 = 5
18 - 32 / 6 * 3 = 3
220 / 5 = 44
27 - 7%3 + 8/3 = 28
  
```

## Task 5:

```

/******************
 * Author: AbdulHadi Afzal
 * Task: Semester GPA Calculation
******************/

#include <stdio.h>

int main() {
    int n, i;
    float totalPoints = 0, totalCredits = 0, gpa, grade, credit;

    printf("Enter number of courses: ");
    scanf("%d", &n);
  
```

```

for (i = 1; i <= n; i++) {
    printf("Enter credit hours for course %d: ", i);
    scanf("%f", &credit);

    printf("Enter grade points (e.g., 4.0, 3.5) for course %d: ", i);
    scanf("%f", &grade);

    totalPoints += credit * grade;
    totalCredits += credit;
}

// Calculate GPA
gpa = totalPoints / totalCredits;
printf("\nYour Semester GPA is: %.2f\n", gpa);

return 0;
}

```

```

Tab@NCDC-0019 MSYS /c/Users/Tab/Desktop/LAB1/Task5
$ gcc -o five five.c
./five
Enter number of courses: 5
Enter credit hours for course 1: 4
Enter grade points (e.g., 4.0, 3.5) for course 1: 4
Enter credit hours for course 2: 3
Enter grade points (e.g., 4.0, 3.5) for course 2: 4
Enter credit hours for course 3: 4
Enter grade points (e.g., 4.0, 3.5) for course 3: 3
Enter credit hours for course 4: 4
Enter grade points (e.g., 4.0, 3.5) for course 4: 3.5
Enter credit hours for course 5: 4
Enter grade points (e.g., 4.0, 3.5) for course 5: 3

Your Semester GPA is: 3.47

```

## Task 6:

```

*****
* Author: AbdulHadi Afzal
* Task: Kirchhoff's Current Law Calculation
*****/

#include <stdio.h>

int main() {
    float V1, R1, R2, R3, i1, i2, i3;

    printf("Enter Voltage (V1): ");
    scanf("%f", &V1);

    printf("Enter Resistance R1: ");
    scanf("%f", &R1);

    printf("Enter Resistance R2: ");
    scanf("%f", &R2);

    printf("Enter Resistance R3: ");
    scanf("%f", &R3);

    // Calculate currents using Ohm's Law and Kirchhoff's Current Law
    i1 = V1 / R1;
    i2 = V1 / R2;
    i3 = i1 - i2;

    printf("\nCalculated Currents:\n");
}

```

```

    printf("i1 = %.2f A\n", i1);
    printf("i2 = %.2f A\n", i2);
    printf("i3 = %.2f A\n", i3);

    return 0;
}

```

```

$ gcc -o six six.c
./six
Enter Voltage (V1): 10
Enter Resistance R1: 5
Enter Resistance R2: 10
Enter Resistance R3: 7

Calculated Currents:
i1 = 2.00 A
i2 = 1.00 A
i3 = 1.00 A

```

## Task 7:

```

/***********************
 * Author: AbdulHadi Afzal
 * Task: Quadratic Equation Extrema Finder
 ***********************/

#include <stdio.h>

// Function to calculate extrema
void find_extrema(double a, double b, double c) {
    if (a == 0) {
        printf("Not a quadratic equation.\n");
        return;
    }

    double x = -b / (2 * a);
    double y = a * x * x + b * x + c;

    printf("Extrema at x = %.2f, y = %.2f\n", x, y);

    if (a > 0)
        printf("This is a Minimum point.\n");
    else
        printf("This is a Maximum point.\n");
}

int main() {
    double a, b, c;

    printf("Enter coefficients a, b, and c: ");
    scanf("%lf %lf %lf", &a, &b, &c);

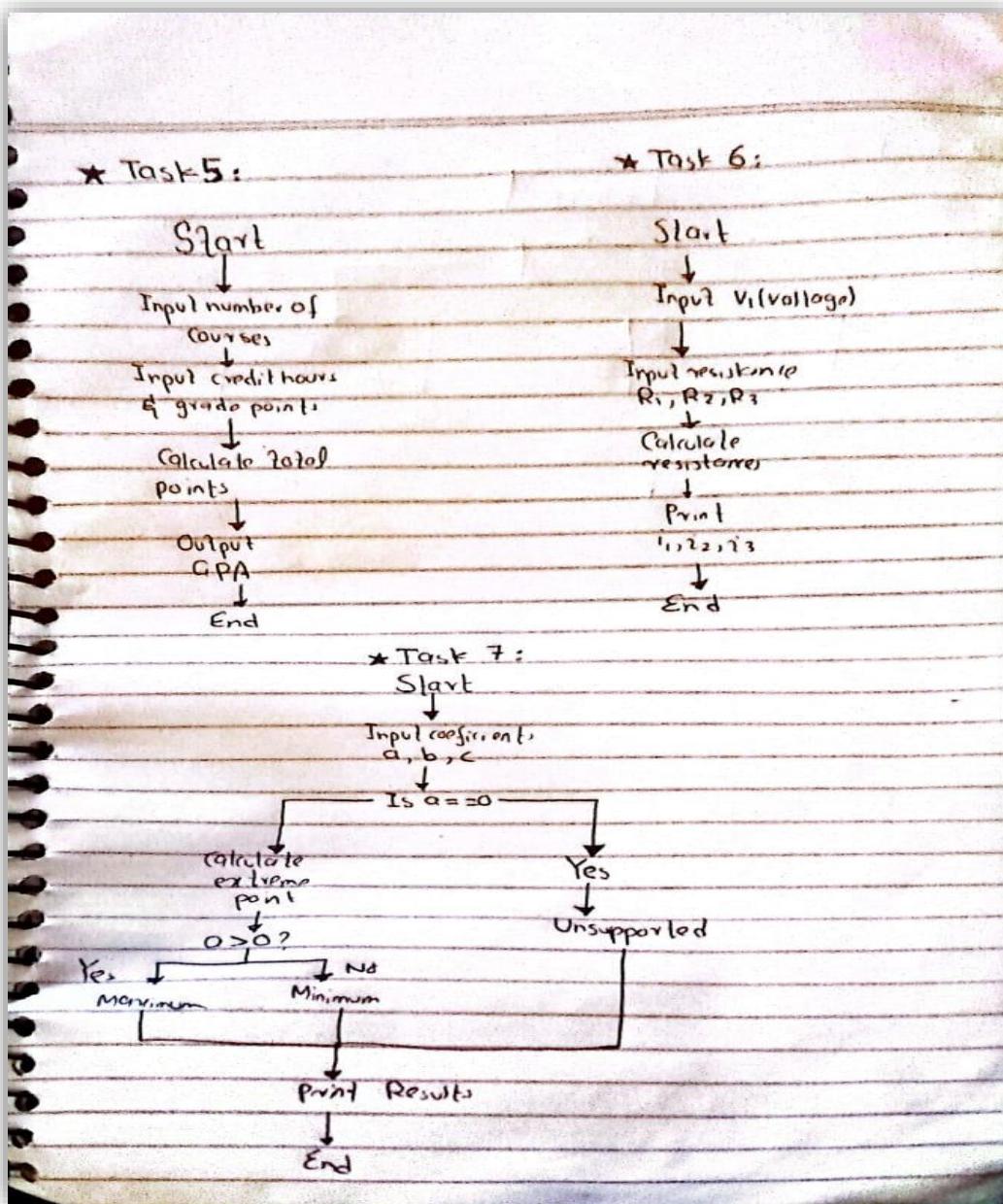
    find_extrema(a, b, c);

    return 0;
}

```

```
# ./sev.exe
Enter coefficients a, b, and c: 1
-2
1
Extrema at x = 1.00, y = 0.00
This is a Minimum point.
```

## Flow Chart:



## Conclusion

This lab introduced essential concepts of compiling, debugging, and working with data types in C. The exercises helped reinforce theoretical concepts with practical applications.