

**CS111 – Introduction to C Programming
(Draft)**

Fall 2025

**Programming Assignment #5 (3%)
Due: Friday October 17, 2025 (23:59)
C Switch and Loops**

Objectives

Further practice the use of control structures in C including **switch**, various **loops**, **break** and **continue** statements.

Part I: Finding prime numbers again

Below is a solution to Part II of Programming Assignment #4. Read and understand this program. Compile and execute it to observe its operation. Then, revise the program so that it reads in two integers, called **low** and **high**, and prints out all the primes between these two integers. In the end, the program should report how many primes are found.

```
#include <stdio.h>

int main() {
    int i, j;
    int prime_count = 0;
    int limit = 5; // Stop after finding 5 prime numbers

    printf("The first %d prime numbers between 1 and 100 are:\n", limit);

    // Outer loop to iterate through numbers 2 to 100
    for (i = 2; i <= 100; i++) {
        // Condition to break the outer loop after finding the required # of primes
        if (prime_count >= limit) {
            break;
        }

        // Assume the number is prime until a divisor is found
        int is_prime = 1;

        // Inner loop to check for divisors
        for (j = 2; j <= i / 2; j++) {
            // If i is divisible by j, it's not a prime
            if (i % j == 0) {
                is_prime = 0; // Set flag to indicate not prime
                break;        // No need to check further divisors, exit inner loop
            }
        }

        // If is_prime is still 1, the number is prime
        if (is_prime) {
            printf("%d ", i);
            prime_count++;
        } else {
            continue;
        }
    }

    printf("\nSearch complete.\n");
    return 0;
}
```

Once the program is completed, add a header documentation section. Include in the header section the answer to the following the question: is the **continue** statement at the end of outer loop necessary? Explain your answer. Submit the program as **pa5p1**.

Part II: Traffic Light Controller

The program below is a simple finite state machine that moves back and forth between two states, S1 and S2, printing out and staying in each for three seconds using the C function **sleep(n)** where n=3 indicating three seconds. Read and understand this program. Modify it so that it simulates a traffic light controller, moving from RED, to GREEN, to YELLOW, and back to RED state, in turn indefinitely. It should stay in RED for seven seconds, GREEN for five and YELLOW for two. Once the program works, name it as **pa5p2.c**, and submit it.

```
#include <stdio.h>
#include <unistd.h> // For the sleep() function

#define S1          1
#define S2          2

int main() {
    short state = S1;

    while (1) {
        printf("State: %d\n", state);
        switch (state) {
            case 1:
                sleep(3); // Stay in State 1 for three seconds
                state = S2; // Transition to State 2
                break;
            case 2:
                sleep(3); // Stay in State 2 three seconds
                state = S1; // Transition to State 1
                break;
        }
    }
    return 0;
}
```

Part III: Largest number finder

Solve Programming Project 1 on p. 122 of the textbook, with the additional requirement that the program should also print out the largest and the smallest number among those that are read. Name your program for this part as **pa5p3.c** and submit it.

Marking and Assignment Submission

This programming assignment is worth a total of 3% with 1% for each part. Submit **pa5p1.c**, **pa5p2.c** and **pa5p3.c** as well as the zipped folder of the three files, called **pa5.zip**, as four separate files through [Blackboard](#). Be prepared to explain to a TA in the next lab session about this assignment.

Notes: Please work on the homework independently. The university has a zero-tolerance policy on plagiarism. Regarding the use of AI to assist you in completing assignments, please refer to “AI Policy” on course Blackboard.