CS111 – Introduction to C Programming
Fall 2025
Programming Assignment #9 (3%)
Due: Friday November 21 (23:59)
C Functions and Pointers

Objectives

In this programming assignment, you will continue to learn how to create C functions and how to use pointer variables.

Part I: Merge function (1%)

Refer to the lecture notes on November 10 where we discuss how *merge sort* works as a recursive C function. In this part of the assignment, you will write a C function (non-recursive) to perform the merging tasking. Specifically, your function is called by the **`main()`** and **`MergeSort()`** functions in the lecture slides so that it completes the final step of merge sort. Its prototype should be:

```
void merge(int arr[], int left, int mid, int right);
```

`merge()` should merge **`arr[]`** from indices **`left`** to **`mid`** with **`arr[]`** from indices **`mid +1`** to **`right`**. For example, if **`left=4`**, **`mid=5`**, and **`right=6`**, then **`merge()`** should merge **`arr[4]`** to **`arr[5]`** with **`arr[6]`**.

Test the operation of your with the provided **`main()`** and **`MergeSort()`**.

Part II: String length function with pointer (1%)

Implement your own **`strlen()`** function named **`my_strlen()`** to calculate the length of a string. Recall that the end of a C string is marked with a NULL character (\0), so that you can use a pointer variable to move along an input string and search for NULL in order to calculate its length. Once **`my_strlen()`** is completed, test it with the following **`main()`** function.

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

int my_strlen(char *);

int main()
{
    // defining string
    char str[] = "Hello, World!";

    // getting length of str using strlen()
    if(strlen(str) == my_strlen(str))
        printf("Success :)\n");
    else
        printf("Failure :(\n");
    return 0;
}
```

Part III: Use of a pointer variable to manipulate an array (1%)

Solve Exercise 17 on p. 275. Test your program with the following `main` function.

```
#define LEN 2

int sum_two_dimensional_array(const int a[][LEN], int);

int main(){
    int N = 3;
    // A 3x2 array
    int arr[][2] = {{1, 2}, {3, 4}, {5, 6}};

    // Call your function
    printf("The sum of the array = ", sum_two_dimensional_array(arr, N);

    return 0;
}
```

Marking and Assignment Submission

Once the program is completed, add a header documentation section and inline comments to explain how the program works. The header section should include information about the author and the general information about this assignment as well as about the use of AI a is described in our "AI policy". Each function should have its own header as the main function to describe the input and output the function and the algorithm used to compute output from the input. (This is called "proper documentation" below.) Name the the programs for the three parts as `pa9p1,` `pa9p2` and `pa9p3.c,` respectively. Submit `pa9p1, pa9p2` and `pa9p3.c` as well as the zipped folder of the three files, called `pa9.zip`, as four separate files through Blackboard. Be prepared to explain to a TA in the next lab session about your solution to this assignment. This programming assignment is worth a total of 3% with 1% for each part.

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.