Tutorial 2: Arrays and Dynamic Arrays

ELEC 278: Fundamentals of Information Structures

The learning goals for Tutorial 2 are:

- Understand the differences between static arrays and dynamic arrays.
- Practice using pointers by creating a dynamic 2D array.

Problem 1. What is the difference between a static array and a dynamic array?

Problem 2. In class, we looked at 1-dimensional arrays. Complete `main()` to create and print a 2-dimensional static array.

```
#include <stdio.h>

int main() {
    // Define the size of the array
    // Feel free to change the size of the 2D array
    int rows = 3;
    int cols = 4;

    // To do: Declare and initialize the 2D array to hold int values

    printf("The 2D array is:\n");
    // To do: Print the array

    return 0;
}
```

Problem 3. We will now create the 2D array and dynamically allocate memory for it on the heap.

Note that for a dynamically allocated 1D array, we create a pointer to a block of memory the size of the array on the heap.

For dynamic 2D arrays, we create a pointer to an array of pointers the size of the rows of the 2D array. This array of (row-)pointers will be on the heap and each of its elements (or pointers) will point to a memory block that is the size of each column.

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

int main() {
    // Feel free to change the size of the 2D array
    int rows = 3, cols = 3;

    // To do: allocate memory for the array of pointers to rows
    int **array = ;
    if (array == NULL) {
        fprintf(stderr, "Memory allocation failed\n");
        return 1;
```

```c
    } // check if memory allocation failed

    // To do: allocate memory for each row
    for (int i = 0; i < rows; i++) {
        // To do
        if (array[i] == NULL) {
            fprintf(stderr, "Memory allocation failed\n");
            return 1;
        }
    } // check if memory allocation failed

    // to do: initialize the array with some values


    // To do: print the array


    // To do: free the allocated memory


    return 0;
}
```