

# CS2050 – Lab 1

## Fall 2023

### Requirements

In this lab, you will cover pass by reference, and using error codes. Each function has an associated error code, and one or more possible error conditions. You should consider when errors might occur that make completion of your function's goal impossible and return the specified error code when they occur.

#### 1.1 `getAverage`

```
int getAverage(int array[], int size, float *result)
```

❶

**Info:** This function takes an array, as well as the size of the array. It calculates the average of the array, if possible, and places the average in the result pointer provided. It returns 0 on success, or 1 on failure.

#### 1.2 `sumPositive`

```
int sumPositive(int array[], int size, int *result)
```

❶

**Info:** This function takes an array, as well as the size of the array. It calculates the sum of the positive numbers in the array, if possible, and places the sum in the result pointer provided. It returns 0 on success, or 1 on failure.

#### 1.3 `sumOdd`

```
int sumOdd(int array[], int size, int *result)
```

❶

**Info:** This function takes an array, as well as the size of the array. It calculates the sum of the odd numbers in the array, if possible, and places the sum in the result pointer provided. It returns 0 on success, or 1 on failure.

### Submission Information

Submit this assignment by the `mucsmake` command.

Use the following submit command on `tc.rnet`:

```
mucsmake <assignment> <filename>
```

For example:

```
mucsmake lab1 lab1.c
```

### Rubric: 9 points

1. Write required `getAverage` function  
\* 3 points
2. Write required `sumPositive` function  
\* 3 points
3. Write required `sumOdd` function  
\* 3 points

### Notice:

1. All of your lab submissions **must** include documentation in the form of code comments to receive full points. In addition, your program is expected to have a **comment header** at the top that includes your name, pawprint, the course you are taking, and the lab that you solved. You can refer to the Lab 0 document for an example of the header.
2. All of your lab submissions must compile under GCC using the `-Wall` and `-Werror` flags to be considered for a grade.
3. Do **NOT** change the given function prototype or anything else in the provided `.h` file.