CS2050 – Lab 1
Spring 2025

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. Consider that parameters passed into your functions may be wrong or illegal (what if I pass in NULL) – you should respond properly to these situations.

1.1 getMedian

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

Info: This function takes an array, as well as the size of the array. It calculates the median (the value for which half the values are greater or equal and half are less or equal) of the array, if possible, and places the median in the result pointer provided. It returns 0 on success, or 1 on failure.

1.2 sumNegative

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

Info: This function takes an array, as well as the size of the array. It calculates the sum of the negative 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 sumEven

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

Info: This function takes an array, as well as the size of the array. It calculates the sum of the even 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. Your TAs can help you with the syntax for mucsmake if needed.

Rubric: 20 points

1. Write required *getMedian* function
   * 8 points
2. Write required *sumNegative* function
   * 6 points
3. Write required *sumEven* function
   * 6 points

Notice:

1. All of your lab submissions **must** include documentation to receive full points.
2. All of your lab submissions must compile under GCC using the *–Wall* and *–Werror* flags to be considered for a grade.
3. You are expected to provide proper documentation in every lab submission, in the form of code comments. For an example of proper lab documentation and a clear description of our expectations, see the lab policy document.