Lab #2

Summer 2023

Requirements

In this lab, you will cover memory allocation, as well as freeing allocated memory. Remember that whenever you allocate memory, you must *always* check that it is not NULL. Make sure to pay attention to the specified return values.

1.1 makeArray

```
int makeArray(int **array, int size)
```

Info: This function takes a pointer to an int array pointer as its first parameter, and the size (i.e. number of elements) of a new array to create as its second parameter. It will create a new array with the given size, and place the address of the newly created array in the provided pointer. It will return **0** if creating the array was successful, or **1** otherwise.

1.2 initArray

```
void initArray(int *array, int size)
```

Info: This function takes an integer array, as well as the size of the array, and initializes the element at each index in the array to ***0***.

1.3 multiplyEven

```
int multiplyEven(int *array, int size, int multiplicand)
```

Info: This function takes an integer array, as well as the size of the array, and multiplies any element of the array which is **even** by the provided multiplicand. It stores the result at the same index in the array, and returns the number of elements which were multiplied. **Note that 0 is considered to be an even number.**

1.4 freeArray

```
void freeArray(int **array)
```

Info: This function takes a pointer to an int array pointer, and frees all memory allocated to that array. After freeing, it also sets the original pointer to NULL.

Submission Information

Submit this assignment by uploading your lab2.c file to Canvas **and** by using the mucsmake command. We are testing the mucsmake command and so we need the lab to be submitted in both ways.

Use the following submit command on tc.rnet:

mucsmake <assignment> <filename>

For example:

mucsmake lab2 lab2.c

Rubric: 12 points

1. Write required *makeArray* function
   * 4 points
2. Write required *initArray* function
   * 2 points
3. Write required *multiplyEven* function
   * 3 points
4. Write required *freeArray* function
   * 3 points

Notice:

5. Do **NOT** change the given function prototype, otherwise your program might not compile properly.
6. All of your lab submissions must compile under GCC using the *–Wall* and *–Werror* flags to be considered for a grade.
7. 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.