Lab #2
Spring 2025

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. Also, consider that you MUST error-check to make sure any parameters passed into your function are valid. For example, will your code respond properly if my test application passes in NULL for the first parameter? Finally, this lab is not about "enhanced" or "smart" arrays – notice that the size parameter is passed into all of the functions that need it.

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 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 each index in the array to ***0***.

1.3 multiplyOdd

```
int multiplyOdd(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 **odd** 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 using the mucsmake command.

Use the following command on Hellbender:

mucsmake <course> <assignment> <filename>

For example:

mucsmake 2050 lab2 lab2.c

Rubric: 20 points

1. Write required *makeArray* function
   * 5 points
2. Write required *initArray* function
   * 5 points
3. Write required *multiplyOdd* function
   * 5 points
4. Write required *freeArray* function
   * 5 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.