Name:

Please remember you must follow instructions exactly in this course. Failure to follow these instructions will result in a failing grade on this quiz.

Do your work in this directory:

/data/courses/ece\_1111/current/quizzes/qu\_09/lastname\_firstname

Make sure your directory permissions are set so that only you can view the contents of the directory.

Copy the code located here:

/data/courses/ece\_1111/current/quizzes/qu\_09/picone\_joseph/

This program resizes an array. It takes two arguments from the command line:

myprog.exe 10 20

The first argument is the size of the array initially created. The second argument is the size of the new array. The array is resized in the function *myresize()*.

Notice that the program works for some arguments:

nedc\_999\_[1]: myprog.exe 3 3

initial array:

data[0] = 0.000000

data[1] = 1.000000

data[2] = 2.000000

good morning class...

new array:

data[0] = -0.000000

data[1] = -1.000000

data[2] = -2.000000

nedc\_999\_[1]: myprog.exe 3 5

initial array:

data[0] = 0.000000

data[1] = 1.000000

data[2] = 2.000000

good morning class...

new array:

data[0] = -0.000000

data[1] = -1.000000

data[2] = -2.000000

data[3] = -3.000000

data[4] = -4.000000

nedc\_999\_[1]: myprog.exe 3 99999

initial array:

data[0] = 0.000000

data[1] = 1.000000

data[2] = 2.000000

good morning class...

Segmentation fault

This is because the *myresize()* function does not do anything. Therefore, at some point the program segfaults because it is accessing memory that hasn’t been allocated.

Change the type declarations of arguments of the *myresize()* function so that the program works AND does not have a memory leak. Do not use global memory or resize the array in the main program. It must be resized in the function.

Do not change the return value of the function – leave it as a Boolean. You can only change the types of the calling arguments.

Also, you must use a float pointer of some sort for data. Do not change the main program. Do not change the data structure to be some sort of C++ class (for those who have experience with more advanced programming languages). You can use the library functions new and delete, but do not use a function such as *realloc()*. I want you to do this simply by changing the type associated with the first argument in *myresize()*, and then adding the corresponding code in *myresize()*. You are essentially writing a function to reallocate memory.

Hint: remember that you cannot change the value of an argument passed to a function in C/C++. Therefore, you must change the value at the memory location at which the pointer data exists in the main program.