Assignment Objective: Build skills on C class creation and integration while implementing a **sorting** utility utilizing the **Heapsort** algorithm.

Requirements:

* Add all new functions at the bottom of the list.cpp file.
* Add the new function definitions to the appropriate places in the list.h file.
* Make a new entry in the Makefile to also build heapSort, utilizing the heapSortMain.cpp file.
* Add the following to your latest list class.
  + Private variable:
    - int heapSize; // tracks the size of the heap that is superimposed over the list
  + Private functions:
    - void buildHeap(); // see lecture notes
    - void heapify(int index);
    - void swap(int &x, int &y);
    - int left(int index) const;
    - int right(int index) const;
  + Public functions:
    - void heapSort(); // see lecture notes
* You must not use any other data structure, whether built-in or otherwise.
* Demonstrate that the sorting works:
  + Run your program as follows:

./heapSort 10 > heapSortOutput.txt

./heapSort 300 >> heapSortOutput.txt

./heapSort 40000 >> heapSortOutput.txt

* + Compare the heapSortOutput.txt to the posted heapSortCorrectOutput.txt file
* Deliverables:
  + Into D2L put a zip file containing files: list.h, list.cpp, Makefile, and heapSortOutput.txt.
  + Turned into class: a hardcopy of your list.h, list.cpp, and heapSortOutput.txt files.