

Program 8 - Building a Heap (Extra Credit)

Due Date: Midnight, December 15, 2016

All programs will be tested on the machines in the Q22 lab. It is required that your code run on the computers in the lab (or via ssh).

*Any changes made to the assignment after posting will be in red
(last updated: 12/5/16 8:18 AM)*

- *There is no supplied driver code for this project.*

Part 1: Heap

- Create an array based **Min-Heap**. You should have a header file, heap.h, with the following:
 - Heap struct containing the following:
 - a Data array
 - as in previous assignment, you should wrap your data in a Data struct.
 - maximum capacity
 - current size
 - A function declaration for a function that allocates a heap, and initializes the internal array to NULL;
- You should also have a source file, heap.c, that implements:
 - Heap * initHeap(Data * d);
 - The function should take a data array, then return a Heap struct with the internal array heapified
- Test your functions and structure to ensure everything is initialized correctly by creating a Heap

Part 2: Heap Operations

- You will need to implement the following functions in your Heap.c file.
- removePriority:
 - removes and returns the priority value from the heap, and re-heapifys.
- You will also need the following for your heap:
 - siftDown
 - heapify

Part 3: Testing Your Heap

- In the driver code, you should have 2 loops:

- The first loops should create an array of 1000 Data objects with random integer values between 1-1000.
 - Once the data Array is created, you should use it to create a heap using your `initHeap()` function.
- Your second loop should remove all elements of the heap by continuously removing the priority value.
 - You must use the [assert](#) library to test your heap to make sure the priority value you remove is greater than or equal to the previous priority value that was removed.

Part 4 - Submission

- Required code organization:
 - `program8.c` - contains the driver code
 - `heap.c/h` - Your header file should have (at minimum) the following function declarations:
 - Data struct
 - `value (int)`
 - Heap struct
 - `data (Data *)`
 - `current_size (int)`
 - `maximum_capacity (int)`
 - `Heap * initHeap(Data * d);`
 - `void siftDown(Heap * h, int index);`
 - `void * heapify(Heap * h);`
 - `Data removePriority(Heap * h);`
 - `makefile`
 - *You must have the following labels in your makefile:*
 - `all` - to compile all your code to an executable called **'program8'** (no extension). **Do not run.**
 - `run` - to compile if necessary and run
 - `checkmem` - to compile and run with valgrind
 - `clean` - to remove all executables and object files
- While inside your program 8 folder, create a zip archive with the following command
 - `zip -r program8 *`
 - This creates an archive of all file and folders in the current directory called `program8.zip`
 - **Do not zip the folder itself, only the files required for the program**
- Upload the archive to Blackboard under Program 8.

Grading Guidelines

Total: 10 points

- **Part 1,2,3:**
 - To get credit, your Heap, and your test loops must all work perfectly.
- **Style Guidelines and Memory Leaks**
 - You will lose significant points for the following:
 - Makefile does not have requested format and labels (-5 points)
 - Does not pass Valgrind Tests (-10 points)
 - Does not follow requested program structure and submission format (-10 points)
 - Does not follow formatting guidelines (-5 points)