

Heap Class Implementation
Cpt S 223 Homework Assignment
Washington State University

In this assignment, you will implement a k-ary max heap template class. You may work under the assumption that whatever type is used for the template type will support the comparison operators, have a copy constructor, and have a proper assignment operator (=). For the actual implementation, you will only ever instantiate the heaps with **int** as the template type.

Like previous assignments, in this assignment you will read simple commands from standard input. You will display your name and ID number on a single line to start with and then enter the command processing loop. There will be one command per line. Command strings will be much like C function calls in terms of syntax. You must support the following commands:

reset(k) : Creates a new k-ary heap using the specified k value. If there is an existing heap, then it is deallocated. There is only one heap active at any given moment in this assignment. The constructor for your heap class should take a k-value and store it in a member variable. This value stays fixed for the lifetime of the heap object. Resetting creates a new heap with 0 items that will be ready for add / remove calls. The k value will always be a valid integer > 1.

- You may assume that the very first command given to your app will be a reset command.
- Note that the reset command can also occur at any point after this.

add(value) : Adds an integer value to the heap. Recall that a heap CAN have duplicates, so you must add this value even if it already exists in the heap.

- The value will be an integer in the range [0, 65535].
- There is no output from this command on screen.

remove() : Removes the max value from the heap AND displays it on its own line without a comma or anything else after it.

displayArray() : Displays the contents of the array that is being used to store the heap. The contents are displayed on one line, in order by array index (0 to n-1). A comma is displayed after each value, including the very last value on the line.

quit() : Breaks the command processing loop and terminates your program. Display the line "Done" as the very last thing before returning from main.

Notes:

- You may use a `std::vector` or just a plain array for the heap storage (with the vector expected to be the easier of the 2).
- See in input/output text files included in the zip file to get an idea of exactly what the commands and the corresponding output look like before you start writing the code.
- Put comments in your code! Give a brief explanation for each function at a minimum.
- Your code must compile with `g++` and run on Linux. Test with I/O redirection using the included input file(s), and use the Linux `diff` utility to compare your output to the expected. Recall that after compilation, the command-line syntax for I/O redirection is:

`./a.out < in.txt > myout.txt`
`diff out.txt myout.txt`

- Look at the sample input/output files in the zip to get an idea of what the commands and their corresponding outputs look like. As usual, develop your own additional test cases on top of these, since they are samples only and not meant to rigorously test your code.