Implement a min-heap of integers as described in class. Remember that if $H$ is the array that contains the heap, the root is $H[1]$. In addition to the standard *insert* and *extractMin* operations, implement the following:

- **Print.** Print the contents of the heap by level. That is, print the root in one line, the children of the root in the next line and so on.
- **Modify ($n,k$).** Set $H[n] = k$ and re-establish the heap property. This method should run in time $O(\log n)$.
- **ExtractSecond.** Remove and return the second smallest element in the heap. This method should run in time $O(\log n)$.
- **ExtractThird.** Remove and return the third smallest element in the heap. This method should run in time $O(\log n)$.
- **ReturnMax.** Returns the largest element without modifying the heap. This method should run in time $O(n)$. Hint: where may the largest element be?.
- **ReturnMax2.** As the one before, but eliminate the need for searching by storing the index of the largest element in a variable. Modify your insertion, extraction, and modification operations to allow your program to keep track of the largest element. Your method should run in time $O(1)$
- **ExtractMax.** As the one before, but you should remove the maximum element from the heap. Your method should run in time $O(\log n)$

As usual, write a report describing your work.