COMP 5112. Lab 4: Heap

Question 1.

The following figure shows a max-heap H, with $H.length = 8$ and $H.size = 4$.

	0	1	2	3	4	5	6	7
H	60	50	40	30				

Insert these numbers (in the following order) into the max-heap H:
11, 88, 22, 77
Draw the content of the heap after each insertion.

Question 2.

Design an algorithm to find the k-th smallest value in an unsorted array $A[0..n-1]$. Your algorithm may use a max-heap H with size k.
Note: you are not allowed to update/sort array A nor create any new array.

As an example, for the array $A[0..9]$ = | 51 | 22 | 73 | 82 | 14 | 67 | 38 | 19 | 98 | 25 |, the 4-th smallest value is 25.

Question 3.

A PriorityQueue is used when the objects are supposed to be processed based on priority. It is known that a Queue follows the First-In-First-Out property, but some applications require the elements to be processed according to the priority. That's when PriorityQueue comes into play.

The PriorityQueue is based on the priority heap. The elements of the priority queue are ordered according to the natural ordering or by a Comparator provided at queue construction time, depending on which constructor is used.

Run MaxHeap.java and MinHeap.java.
What are the output results?

Question 4.

Given two integer arrays, *U* and *V* sorted in **ascending order** and an integer *k*.
Let (*u*, *v*) be a pair such that *u* is from array *U* and *v* is from array *V*.
Please try to write a program that returns the *k* pairs (u_1, v_1), (u_2, v_2), ..., (u_k, v_k) with **the smallest sums**.

Example 1:
Input: *U* = [1,7,11], *V* = [2,4,6], *k* = 3
Output: [[1,2],[1,4],[1,6]]
Explanation: The first 3 pairs are returned from the sequence:
[1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]

Example 2:
Input: *U* = [1,1,2], *V* = [1,2,3], *k* = 2
Output: [[1,1],[1,1]]
Explanation: The first 2 pairs are returned from the sequence:
[1,1],[1,1],[1,2],[2,1],[1,2],[2,2],[1,3],[1,3],[2,3]