Problem 1. (*Certify Heap*) Implement the static method `isMaxHeap()` in `CertifyHeap.java` that takes an array `a` of `Comparable` objects (excluding `a[0] = *`) and returns `true` if `a` represents a max-heap, and `false` otherwise.

```
>_ ~/workspace/exercise4

$ java CertifyHeap
* M A X H E A P
<ctrl-d>
false
$ java CertifyHeap
<ctrl-d>
* A A E H M P X
false
$ java CertifyHeap
<ctrl-d>
* X P M H E A A
true
```

Problem 2. (*Ramanujan's Taxi*) Srinivasa Ramanujan was an Indian mathematician who became famous for his intuition for numbers. When the English mathematician G. H. Hardy came to visit him one day, Hardy remarked that the number of his taxi was 1729, a rather dull number. To which Ramanujan replied, "No, Hardy! It is a very interesting number. It is the smallest number expressible as the sum of two cubes in two different ways." Verify this claim by writing a program `Ramanujan1.java` that accepts n (int) as command-line argument and writes to standard output all integers less than or equal to n that can be expressed as the sum of two cubes in two different ways. In other words, find distinct positive integers a, b, c, and d such that $a^3 + b^3 = c^3 + d^3 \leq n$.

```
>_ ~/workspace/exercise4

$ java Ramanujan1 10000
1729 = 1^3 + 12^3 = 9^3 + 10^3
4104 = 2^3 + 16^3 = 9^3 + 15^3
```

Directions:

- Use four nested `for` loops, with these bounds on the loop variables: $0 < a \leq \sqrt[3]{n}$, $a < b \leq \sqrt[3]{n - a^3}$, $a < c \leq \sqrt[3]{n}$, and $c < d \leq \sqrt[3]{n - c^3}$

Do not explicitly compute cube roots, and instead use `x * x * x < y` in place of `x < Math.cbrt(y)`.

Problem 3. (*Ramanujan's Taxi Redux*) Write a program `Ramanujan2.java` that uses a minimum-oriented priority queue to solve the problem from Exercise 2.

```
>_ ~/workspace/exercise4

$ java Ramanujan2 10000
1729 = 1^3 + 12^3 = 9^3 + 10^3
4104 = 9^3 + 15^3 = 2^3 + 16^3
```

Directions:

- Initialize a min-PQ `pq` with pairs $(1, 2), (2, 3), (3, 4), \ldots, (i, i + 1)$, where $i < \sqrt[3]{n}$
- Declare `Pair` objects `prev` and `curr`
- While `pq` is not empty:
  - Set `prev` to `curr` and `curr` to `pq.delMin()`
  - Print the previous pair (k, l) and current pair (i, j) if $k^3 + l^3 = i^3 + j^3 \leq n$
  - If $j < \sqrt[3]{n}$, insert the pair $(i, j + 1)$ into `pq`

Again, do not explicitly compute cube roots, and instead use `x * x * x < y` in place of `x < Math.cbrt(y)`.

Files to Submit

1. `CertifyHeap.java`
2. `Ramanujan1.java`
3. `Ramanujan2.java`

Before you submit your files, make sure:

- You do not use concepts outside of what has been taught in class.
- Your code is adequately commented, follows good programming principles, and meets any specific requirements such as corner cases and running times.