Devil Number

In this problem you are to complete four methods in the `DevilNumbers` class. The four methods are `isDevilNumber`, `getLargestDevilNumber`, `isTrueDevilNumber` and `getLargestTrueDevilNumber`.

A positive number is a Devil Number if any combination of the individual digits making up the number sum to 6. For example, 472 (4 + 2 = 6) and 463 (by default, any number containing a 6 is a Devil Number) are Devil numbers. Since no subset of the digits of 714 can be found that sums to 6, 714 and 471 are not Devil Numbers. The `isDevilNumber(num)` returns true if its `int` parameter `num` is a Devil Number and returns false otherwise.

The following tables show sample results of the `isDevilNumber(int num)` method.
You may assume `num > 0`

The following code	Returns
`DevilNumbers.isDevilNumber(2)`	`false`
`DevilNumbers.isDevilNumber(4305)`	`false`
`DevilNumbers.isDevilNumber(714)`	`false`
`DevilNumbers.isDevilNumber(471)`	`false`
`DevilNumbers.isDevilNumber(18047)`	`false`
`DevilNumbers.isDevilNumber(52370)`	`false`
`DevilNumbers.isDevilNumber(76)`	`true`
`DevilNumbers.isDevilNumber(472)`	`true`
`DevilNumbers.isDevilNumber(111111)`	`true`
`DevilNumbers.isDevilNumber(207060)`	`true`
`DevilNumbers.isDevilNumber(273021)`	`true`
`DevilNumbers.isDevilNumber(7152021)`	`true`

The `getLargestDevilNumber(num)` returns the largest Devil Number less than or equal to its `int` parameter `num`. If no Devil Number exists, return -1.

The table on the following page show sample results of the `getLargestDevilNumber(num)` method.

The following tables show sample results of the `getLargestDevilNumber (int num)` method.
You may assume `num > 0`

The following code	Returns
`DevilNumbers.getLargestDevilNumber(5)`	`-1`
`DevilNumbers.getLargestDevilNumber(720310)`	`720310`
`DevilNumbers.getLargestDevilNumber(43095)`	`43093`
`DevilNumbers.getLargestDevilNumber(1040)`	`1036`

A True Devil Number is a Devil Number which contains no 7. For example, 452 and 643 are Devil numbers, do not contain a 7 and are therefore True Devil Numbers. While 472 and 76 are both Devil Numbers, both are not True Devil Numbers because they contain one (or more) 7s. The `isTrueDevilNumbr(num)` returns true if its `int` parameter `num` is a True Devil Number and returns false otherwise.

The following tables show sample results of the `isTrueDevilNumber(num)` method.
You may assume `num > 0`

The following code	Returns
`DevilNumbers.isDevilNumber(472)`	`false`
`DevilNumbers.isDevilNumber(76)`	`false`
`DevilNumbers.isDevilNumber(720310)`	`false`
`DevilNumbers.isDevilNumber(43027)`	`false`
`DevilNumbers.isDevilNumber(10471)`	`false`
`DevilNumbers.isDevilNumber(52370)`	`false`
`DevilNumbers.isDevilNumber(452)`	`true`
`DevilNumbers.isDevilNumber(643)`	`true`
`DevilNumbers.isDevilNumber(111111)`	`true`
`DevilNumbers.isDevilNumber(20060)`	`true`
`DevilNumbers.isDevilNumber(213021)`	`true`
`DevilNumbers.isDevilNumber(9152021)`	`true`

The `getLargestDevilNumber(num)` returns the largest Devil Number less than or equal to its `int` parameter `num`. If no Devil Number exists, return -1.

The following tables show sample results of the `getLargestTrueDevilNumber(int num)` method. You may assume `num > 0`

The following code	Returns
`DevilNumbers.getLargestTrueDevilNumber(4)`	-1
`DevilNumbers.getLargestTrueDevilNumber(111110)`	111106
`DevilNumbers.getLargestTrueDevilNumber(7060)`	6999
`DevilNumbers.getLargestTrueDevilNumber(217819)`	216999