10-happy numbers

Happy number - Wikipedia https://en.wikipedia.org/wiki/Happy_number

In number theory, a -happy number is a natural number in a given number base that eventually reaches 1 when iterated over the perfect digital invariant function for =. Those numbers that do not end in 1 are -unhappy numbers (or -sad numbers). The origin of happy numbers is not clear. Happy numbers were brought to the attention of Reg Allenby (a British author and senior lecturer in pure ... (not important to this problem)

In number theory, a **perfect digital invariant (PDI)** is a number in a given number base that is the sum of its own digits each raised to a given power p.

This problem will assume number base 10 and given power p = 2.

For example, 19 is 10-happy since: $19 \rightarrow 1^2 + 9^2 = 82 \rightarrow 8^2 + 2^2 = 68 \rightarrow 6^2 + 8^2 = 100 \rightarrow 1^2 + 0^2 + 0^2 = 1$. Note the sequence includes the number 19, 82, 68, 100 and 1. Excluding the final value of 1, the length of the sequence is 4.

In addition, 989 is 10- happy since: $989 \rightarrow 9^2 + 8^2 + 9^2 = 226 \rightarrow 2^2 + 2^2 + 3^2 = 44 \rightarrow 4^2 + 4^2 = 32 \rightarrow 3^2 + 2^2 = 13 \rightarrow 1^2 + 3^2 = 10 \rightarrow 1^2 + 0^2 = 1$. Note the sequence includes the number 989, 226, 44, 32, 13, 10 and 1. Excluding the final value of 1, the length of the sequence is 6.

The number 999 is not 10-happy since: $999 \rightarrow 9^2 + 9^2 + 9^2 = 243 \rightarrow 2^2 + 4^2 + 3^2 = 29 \rightarrow 2^2 + 9^2 = 85 \rightarrow 8^2 + 5^2 = 89 \rightarrow 8^2 + 9^2 = 145 \rightarrow 1^2 + 4^2 + 5^2 = 42 \rightarrow 4^2 + 2^2 = 20 \rightarrow 2^2 + 0^2 = 4 \rightarrow 4^2 = 16 \rightarrow 1^2 + 6^2 = 37 \rightarrow 3^2 + 7^2 = 58 \rightarrow 5^2 + 8^2 = 89$. At this point a cycle has formed (89, 145, 42, 20, 4, 16, 37, 58, 89) making this number 10-unhappy.

The distinct combinations of digits that form 10-happy numbers below 1000 are (the rest are just rearrangements and/or insertions of zero digits):

> 1, 7, 13, 19, 23, 28, 44, 49, 68, 79, 129, 133, 139, 167, 188, 226, 236, 239, 338, 356, 367, 368, 379, 446, 469, 478, 556, 566, 888, 899

The first pair of consecutive 10-happy numbers is 31 and 32. The first set of three consecutive is 1880, 1881, and 1882. It has been proved that there exist sequences of consecutive happy numbers of any natural-number length. The beginning of the first run of at least *n* consecutive 10-happy numbers for *n* = 1, 2, 3, ... is

> 1, 31, 1880, 7839, 9, 7899999999999959999999996, 7899999999999959999999996

You will implement two static methods in the `HappyNumbers` class. The two methods you will implement are the `getHappyLength(int num)`, and the `makeHappy(int[] num)` methods.

The `getHappyLength(int num)` returns the length of the sequence in determining if the parameter `num` is 10-happy. Remember, the length does **NOT** include the final ending value of 1. If the parameter `num` is **NOT** 10-happy return -1.
The following code shows the results of the `getHappyLength` method.

The following code	Returns
`HappyNumbers.getHappyLength(19);`	`4`
`HappyNumbers.getHappyLength(44);`	`4`
`HappyNumbers.getHappyLength(921);`	`3`
`HappyNumbers.getHappyLength(989);`	`6`
`HappyNumbers.getHappyLength(999)`	`-1`

The `makeHappy(int num)` returns the smallest of all numbers surrounding the parameter `num` that is 10-happy, or -1 if none of the surrounding numbers are 10-happy. You may assume the parameter `num` is **NOT** 10-happy. A number is said to surround another number if exactly one digit is within one of the digit in the original number. For example, the number surrounding:

- 45 are: 35, 55, 44 and 46, with 44 being the smallest (and only) 10-happy number
- 2503 are: 1503, 3503, 2403, 2603, 2593, 2513, 2502, and 2504, with 2603 being the only 10-happy number.
- 612 are: 512, 712, 602, 622, 611, and 613, with 622 being the only 10-happy number
- 64 are 54: 54, 74, 63 and 65. None of these numbers are 10-happy and the method returns -1.

It should be noted the digits surrounding the digit 9 are 8 and 10. For example, the numbers surrounding:

- 9 are: 8 and 10, with 10 being the only 10-happy number.
- 999 are: 899, 1099, 989, 9109, 998, and 9910, with 899, 989 and 998 being 10-happy numbers, of which 899 is the smallest.

The one exception is for number with a leading 1. When the leading digit is a 1, the two surrounding digits are 9 and 2. For example, the numbers surrounding:

- 123 are: 223, 923, 113, 133, 122, and 124, with 923 and 133 being the only 10-happy numbers with 133 being the smallest
- 189 are: 289, 989, 199, 179, 188, and 1810, with 989 and 188 being the only 10-happy numbers with 188 being the smallest

If there does NOT exist a 10-Happy number surrounding the parameter `num`, return -1.

The following code shows the results of the `makeHappy` method.

The following code	Returns
`HappyNumbers.makeHappy(45);`	`44`
`HappyNumbers.makeHappy(2503);`	`2603`
`HappyNumbers.makeHappy(612));`	`622`
`HappyNumbers.makeHappy(64));`	`-1`
`HappyNumbers.makeHappy(9);`	`10`
`HappyNumbers.makeHappy(999);`	`899`
`HappyNumbers.makeHappy(123);`	`133`
`HappyNumbers.makeHappy(189);`	`188`