A number has the property of being *DoubleMinded* if it contains exactly one pair of identical digits, and all other digits appear at most once in the number.

The following numbers are examples of numbers with the *DoubleMinded* property:

* 66
* 26964
* 1036850
* 112034

The following numbers are examples of numbers that do **NOT** have the *DoubleMinded* property:

* 8 // no pair of identical digits
* 2964 // no pair of identical digits
* 1003650 // three identical digits (0 appears three times)
* 16861 // two pair of identical digits (two 6’s and two 1’s)

// or one pair of identical digits, another digits appears more thanonce

In this problem you are to implement three static methods in the DoubleMinded class. The first method to implement is isDoubleMindedNumber(int num). isDoubleMindedNumber returns true if the parameter num has the *DoubleMinded* property.

The following code shows the results of the isDoubleMindedNumber method.

|  |  |
| --- | --- |
| The following code | Returns |
| DoubleMindedNumbers.isDoubleMindedNumber(66) | true |
| DoubleMindedNumbers.isDoubleMindedNumber(26964) | true |
| DoubleMindedNumbers.isDoubleMindedNumber(1036850) | true |
| DoubleMindedNumbers.isDoubleMindedNumber(112034) | true |
| DoubleMindedNumbers.isDoubleMindedNumber(8) | false |
| DoubleMindedNumbers.isDoubleMindedNumber(2964) | false |
| DoubleMindedNumbers.isDoubleMindedNumber(1003650) | false |
| DoubleMindedNumbers.isDoubleMindedNumber(16861) | false |

The second method to implement is the distanceToNextDoubleMindedNumber(int num). distanceToNextDoubleMindedNumber returns the minimum (non negative) value d, such that the sum, d + the parameter num, has the *DoubleMinded* property.

The following code shows the results of the distanceToNextDoubleMindedNumber method.

|  |  |
| --- | --- |
| The following code | Returns |
| DoubleMindedNumbers.distanceToNextDoubleMindedNumber(8) | 3 |
| DoubleMindedNumbers.distanceToNextDoubleMindedNumber(295) | 4 |
| DoubleMindedNumbers.distanceToNextDoubleMindedNumber(66) | 0 |
| DoubleMindedNumbers.distanceToNextDoubleMindedNumber(111261) | 773 |

The third method to implement is the getDoubleMindedBetween(int min, int max). getDoubleMindedbetween(int min, int max) returns an array containing all numbers with the *DoubleMinded* property between the parameters min and max, inclusive.

The following code shows the results of the getDoubleMindedBetween method.

|  |  |
| --- | --- |
| The following code | Returns |
| int[] ans = DoubleMindedNumbers.getDoubleMindedBetween(34, 65) |  |
| ans.length | 2 |
| ans[0] == 44 || ans[1] == 44 | true |
| ans[0] == 55 || ans[1] == 55 | true |

The following code shows the results of the getDoubleMindedBetween method.

|  |  |
| --- | --- |
| The following code | Returns |
| int[] ans = DoubleMindedNumbers.getDoubleMindedBetween(121, 131) |  |
| ans.length | 3 |
| ans[0] == 121 || ans[1] == 121 || ans[2] == 121 | true |
| ans[0] == 122 || ans[1] == 122 || ans[2] == 122 | true |
| ans[0] == 131 || ans[1] == 131 || ans[2] == 131 | true |