Partition This

All integers can be partitioned into groups (without rearranging the order of the digits) in such a way that the sum of the individual digits of each group is equal to the sum the individual digits of all the other groups. For example, we say the number $614,377$ is Partitionable with degree 7 since it can be partition in 4 groups $[61,43,7,7]$, each with the sum of the digits equal to 7.
Note: For all numbers in this problem, ignore sign. That is use the absolute value of all numbers for your calculations.

In this problem you are to complete three static methods in the `PartitionThis` class. The three methods are `sumDigits`, `isPartitionable` and `minPartitionDegree`.

The `SumDigits(int num)` method returns the sum of the individual digits of its argument `num`. For example, `SumDigits(103)` returns 4 since $4 = 1 + 0 + 3$ and `SumDigits(-24)` returns 6 since $6 = 2 + 4$.

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

The following code	Returns
`PartionThis.sumDigits(103)`	4
`PartionThis.sumDigits(-24)`	2+4 = 6
`PartionThis.sumDigits(15086)`	1+5+0+8+6 = 20
`PartionThis.sumDigits(-9237)`	9+2+3+7 = 21

The `isPartitionable(int number, int deg)` method returns true if the `number` can be partitioned with degree `deg`. For example, `isPartitionable(415041131, 5)` returns true since `415041131` can be partitioned as $= 41, 50, 41$ and 131 and `isPartitionable(415041132, 5)` returns false since `415041132` can NOT be partitioned into groups in such a way that the sum of the individual digits of each group is 5.
Remember, you may NOT rearrange the order of the digits in the number.

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

The following code	Returns
`PartionThis.isPartionable(415041131, 5)`	true
`PartionThis.isPartionable(410250101, 7)`	true
`PartionThis.isPartionable(415041132, 5)`	false

The `minPartionDegree(int num)` method returns the minimum value `deg` such that the degree of the partition of `num` is `deg`. Recall that all integers are partitionable by the sum of its digits. For example, `minPartionDegree (415041131)` returns 5 since `415041131` can be partitioned with degree 5, but cannot be partitioned with degree 0, 1, 2, 3 or 4. And `minPartitionDegree(3054628)` returns 28 since `3054628` can NOT be partitioned into groups in such a way that the sum of the individual digits of each group is any value.

Partition This

The only partition of `3054628` is as a single number, and `minPartionDegree(3054628)` returns 3+0+5+4+6+2+8 = 28.

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

The following code	Returns
`PartionThis.minPartionDegree(415041131)`	5
`PartionThis.minPartionDegree(3054628)`	28
`PartionThis.minPartionDegree(0)`	0