# Cool Numbers

You will implement four static methods in the `CoolNumbers` class. The four methods you will implement are the `getDensce(int num)`, the `getShavar(int num)`, the `getCoolness(int num)`, and the `getNextNumberWithGreaterCoolness(int num)` methods.

Note: This problem requires the converting of decimal numbers to their binary representation and you may use the existing static method `Integer.toBinaryString(int num)` which returns a string representation of the integer argument (`num`) as an unsigned integer in base 2:

```
    String s = Integer.toBinaryString(num);
```

The `getDensce(int num)` method returns the larger of 0 or 2 less than the length of the longest consecutive run of 1s in its binary representation.

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

| The following code | Returns |
|---|---|
| `CoolNumbers.getDensce(7)       //  7 = 111`$_2$ | 1 |
| `CoolNumbers.getDensce(316)     //  316 = 100111100`$_2$ | 2 |
| `CoolNumbers.getDensce(886)     //  886 = 1101110110`$_2$ | 1 |
| `CoolNumbers.getDensce(14319)   //  14319  = 11011111101111`$_2$ | 4 |
| `CoolNumbers.getDensce(27867)   //  27867 = 110110011011011`$_2$ | 0 |
| `CoolNumbers.getDensce(16)      //  16 = 10000`$_2$ | 0 |

The `getShavar(int num)` method counts the number of times the String "101" occurs in the binary representation of the parameter `num`. Every occurrence increases the return value by 2, unless the first 1 is being shared (e.g., `21 = "10101"`), then the return value is increased by three (e.g. `"10101"` gives 2 for the first `"101"`, and the second `"101"` gives 1 for a total of 3).

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

| The following code | Returns |
|---|---|
| `CoolNumbers.getShavar(21)   //  21 = 10101`$_2$ | 3 |
| `CoolNumbers.getShavar(45)   //  45 = 101101`$_2$ | 4 |
| `CoolNumbers.getShavar(429)  //  429 = 110101101`$_2$ | 5 |
| `CoolNumbers.getShavar(5461) //  5461 = 1010101010101`$_2$ | 7 |
| `CoolNumbers.getShavar(85)   //    85 = 1010101`$_2$ | 4 |
| `CoolNumbers.getShavar(725)  //   725 = 1011010101`$_2$ | 6 |

The `getCoolness(int num)` method returns the sum of `getDensce` score and `getShavar` rating

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

| The following code | Returns |
|---|---|
| `CoolNumbers.getCoolness(117)    //   117 = 1110101` | `4 = 1 + 3` |
| `CoolNumbers.getCoolness(99)     //    99 = 1100011` | `0` |
| `CoolNumbers.getCoolness(6101)   //  6101 =` $1011111010101_2$ | `9 = 3 + 6` |
| `CoolNumbers.getCoolness(14319)  //  14319  =` $11011111101111_2$ | `8 = 4 + 4` |

The `getNextNumberWithGreaterCoolness(num)` returns the next integer with a `getCoolness` greater than `getCoolness(num)` the parameter. That is, if:

```
      ans = CoolNumbers.getNextNumberWithGreaterCoolness(num)
```

then

```
      CoolNumbers.getCoolness(k) < CoolNumbers.getCoolness(ans), num <= k < ans
```

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

| The following code | Returns |
|---|---|
| `CoolNumbers.getNextNumberWithGreaterCoolness(99)` | `101` |
| `CoolNumbers.getNextNumberWithGreaterCoolness(101)` | `106` |
| `CoolNumbers.getNextNumberWithGreaterCoolness(253)` | `381` |