Schooner Dice

Intro:

We'd like to get a sample of your coding style. We're most interested in your code structure, process and thinking. Please implement the problem in whatever language you feel most comfortable with, and be sure to include instructions on how to compile and run your sample. The exercise should take no longer than 2-3 hours.

Requirements:

The module we'd like you to produce should provide two methods to help power the scoring system of a poker based dice game. The game consists of rolling five 8-sided dice and determining a score based on the table below. The first method will produce a score to display in the scorecard. The second will show the highest scoring categories to help players determine the best score for their roll.

```
// returns the score of the dice for the specified category
 int score(Enum category, List<int> diceRoll)

// returns the best scoring category of all qualifying categories, or a list if there is a tie for best category
List<Enum> topCategories(List<int> diceRoll)

Category Enum:
ONES, TWOS, THREES, FOURS, FIVES, SIXES, SEVENS, EIGHTS, THREE_OF_A_KIND, FOUR_OF_A_KIND, FULL_HOUSE,
SMALL_STRAIGHT, ALL_DIFFERENT, LARGE_STRAIGHT, SCHOONER, CHANCE
```

- The dice roll list contains five ints, each representing the face value of a die
    - Face values range from 1 to 8 inclusive.
- You are guaranteed to have the proper number of dice and no invalid values.
- In the score method, if the diceRoll doesn't qualify for a category, the method should return a (0).

Category	**Qualifies When...**	**Score**
ONES, TWOS, THREES, FOURS, FIVES, SIXES, SEVENS, EIGHTS	Any combination	The sum of all dice of the selected number
THREE_OF_A_KIND	At least three dice the same	Sum of all dice
FOUR_OF_A_KIND	At least four dice the same	Sum of all dice
FULL_HOUSE	Three of one number and two of another	25
SMALL_STRAIGHT	Four sequential dice	30
ALL_DIFFERENT	No duplicate numbers	35
LARGE_STRAIGHT	Five sequential dice	40
SCHOONER	All dice the same	50
CHANCE	Any combination	Sum of all dice

Examples:

```
// Input to 'score' method
(FULLHOUSE, [1, 1, 1, 7, 7])

// Expected output (Full House)
25



// Input to ‘topCategories’ method
[3, 3, 3, 6, 7]

// Expected output
[THREE_OF_A_KIND, CHANCE]
```