1. We have chosen the array representation of a polynomial: RatNum[] coeffs, where coeffs[i] stores the coefficient of the term of exponent i. An alternative data representation is the list-of-terms representation: List<Term> terms, where each Term object stores the term's RatNum coefficient and integer exponent. The beauty of the ADT methodology is that we can switch from one representation to the other without affecting the clients of our RatPoly. Briefly list the advantages and disadvantages of the array representation versus the list-of-terms representation.

   By using a list, we can access the object easily and adding and removing will be really convenient. The disadvantage is the size will be larger and run time will be longer since it's always increasing the size while we add things.

   By using a list, we can access the coefficient directly by the index(which is also the exponent). Also, it's easier to deal with fraction. However, it only has a limited size. Every time we need to add stuff, we need to create a new one.

2. Where did you include calls to checkRep in RatPoly (at the beginning of methods, the end of methods, the beginning of constructors, the end of constructors, some combination)?

   We need to check at the end of the constructor. Because that's where the object is created. RatPoly is immutable after created. So, we need to check the representation invariant at the end.