**String**

**Zigzag Conversion** (**6**)

1. **Problem**: Given string, return string after zigzag conversion;
2. **Solution**: Time Complexity should be O(n); Space Complexity should be O(1).
   1. Use jump pattern to avoid additional space;
   2. Traverse by row, calculate the jump.

**String to Integer (atoi)** (**8**)

1. **Problem**:
2. **Solution**: Easy one

**Longest Common Prefix** (**14**)

1. **Problem**: Given array, return the longest common prefix string;
2. **Solution**: Easy one