一、穷举iOS下所有的本地持久化方案。  
答：所有的本地持久化数据存储的本质都是写文件，而且只能存到沙盒中。

沙盒机制是苹果的一项安全机制，本质就是系统给每个应用分配了一个文件夹来存储数据，而且每个应用只能访问分配给自己的那个文件夹，其他应用的文件夹是不能访问的。  
数据存储的核心都是写文件。主要有四种持久化方式：属性列表，对象序列化，SQLite 数据库, CoreData

# 二、viewWillLayoutSubView是什么？

答：当viewController的bounds又改变，调用这个方法来实现subview的位置。可重写这个方法来实现父视图变化subview跟着变化。

# 三、UIView和CALayer是什么关系?

**答：**能答出UIView是CALayer的delegate就及格了，能说出UIView主要处理事件，CALayer负责绘制就更好，再聊下二者在使用过程中对动画流畅性影响的注意点就更好了，UI流畅性是个大话题。

一个UIView默认会有一个layer，当然一个UIView可以有很多个layer。UIView负责处理事件，而CALayer负责绘制，而绘制渲染又分为离屏渲染和当前屏渲染。

# 四、说一下UITableViewCell的卡顿你是怎么优化的？

答：1.提前计算并缓存好高度，因为heightForRow最频繁的调用。  
2.异步绘制，遇到复杂界面，性能瓶颈时，可能是突破口。  
3.滑动时按需加载，这个在大量图片展示，网络加载时，很管用。（SDWebImage已经实现异步加载）。  
4.重用cells。  
5.如果cell内显示得内容来自web，使用异步加载，缓存结果请求。  
6.少用或不用透明图层，使用不透明视图。  
7.尽量使所有的view opaque，包括cell本身。  
8.减少subViews  
9.少用addView给cell动态添加view，可以初始化的时候就添加，然后通过hide控制是否显示。

# 五、.简单解释一下Instrument，并且阐述你用Instrument优化动画性能的经历。

答：Xcode自带的检测调试工具Instruments。

Instruments提供了很多功能，我会重点介绍一下我最常用的几类：

1.Time Profiler：CPU分析工具分析代码的执行时间。

2.Core Animation：离屏渲染，图层混合等GPU耗时。

3.Leaks：内存检测，内存泄漏检测工具。

4.Energy Log：耗电检测工具。

5.Network：流量检测工具。