1. What is the difference in recyclerView and listView?

ListVIew	RecyclerView
Native	Library must be implemented (3rd party libs) The RecyclerView widget is a more advanced and flexible version of ListView
Eager Loading	Lazy loading
Use ArrayAdapter	Use adapter RecyclerVIew.Adapter
ItemClick Listener Built in	Item ClickListener must be implemented
Just work Vertical	Can work: vertical-Horizontal-grid
Don’t have layout Manager	Have a Layout Manager
Older	Newer
Does not enforce view Holder pattern	Must use ViewHolder Pattern
	It’s most efficient than ListView

2. Define lazy loading (also called on-demand loading).

Lazy loading is a design pattern commonly used in computer programming to defer initialization of an object until the point at which it is needed. It can contribute to efficiency in the program's operation if properly and appropriately used. The opposite of lazy loading is eager loading

3. What is an item decorator in RecyclerViews?

It is a tool used to decorate the children of a RecyclerView. With ItemDecoration you can easily modify the appearance of these child views. It’s used for replace the property divide in ListView

4. What is the View Holder Pattern?

The ViewHolder design pattern enables you to access each list item view without the need for the look up, saving valuable processor cycles. Specifically, it avoids frequent call of findViewById() during ListView scrolling, and that will make it smooth.

5. How do you implement an item touch helper for the RecyclerView?

ItemTouchHelper is an Android utility class that adds support for touchscreen gestures to the RecyclerView.

1. We need to create a new interface *ItemTouchHelperAdapter* this one can be created in a package utils with the following two methods to override
    a. Method **onItemMove()** will be called each time the user moves an item by dragging it across the touch screen. The argument *fromPosition* represents the location the item originally resided at. *toPosition* represents the location the user moved the item to.
    b. Method **onItemDismiss()** is called when an item has been dismissed with a swipe motion. The parameter position represents the location of the dismissed item.
2. we'll create another interface in the util subpackage with the name OnStartDragListener with the following method to override
    a. Method **onStartDrag()** will be called when the user begins a 'drag-and-drop' interaction with the touchscreen. *viewHolder* represents the RecyclerView holder corresponding to the object being dragged.