Observations:

Task 1: In task 1 the APK files for the Bitmoji Android app was downloaded. This task showed how easy it was to obtain APK files for Android apps to perform the repackaging attack.

Task 2: In task 2 we used APKTool to disassemble the Bitmoji dex code to smali code. In this task it is easy to see that if you have APKTool it is easy to disassemble any APK files for Android apps.

Task 3: In task 3 we inserted a MaliciousCode.smali file in the smali/com folder and then edited the AndroidManifest.xml file so that the system knows when to invoke the broadcast receiver. In this task we see that when the application is disassembled it is easy to insert whatever malicious code we would want to write and for this task we specifically target the broadcast receiver.

Task 4: Since we have the malicious code is inserted now we have to repackage the application with the APKTool. Once the application is repackaged we have to sign the APK file so generated the public and private key using the keytool command. Then jarsigner is used to sign the APK file using the keys that were generated. In this task we see that even though we are the ones inserting the malicious code we still need to make sure no one else can get to the code we inserted or the purpose of inserting it would be pointless.

Task 5: The last part of the lab has us install the Bitmoji app with the malicious code onto the Android VM and see if the attack works. With this task we see that our malicious code is successful in deleting the contacts in the VM.

Question 1: Why is the repackaging attack not much a risk in iOS devices?
Answer:

- Because on iOS devices, the App Store does not allow unverified publishers to add applications, because of this there are generally not repackaged apps on iOS, as the publisher would have to be registered and approved with Apple.

Question 2: If you were Google, what decisions you would make to reduce the attacking chances of repackaging attacks?
Answer:

- I would attempt to make the apps harder to publish onto the Google Play store, much like Apple, I would go through a process of verification for developers who want to upload content. Or maybe have some sort of icon showing users that a developer is verified, and that apps without the icon come at an inherent risk.

Question 3: Third-party markets are considered as the major source of repackaged applications. Do you think that using the official Google Play Store only can totally keep you away from the attacks? Why or why not?
Answer:

- No, Google Play Store cannot totally keep a customer safe from the attacks, much in the same way that antivirus softwares are not always 100% effective. Even if Google was to deploy a built-in malware scanner, it would have to be updated constantly in order to keep up with the newest malware.

Question 4: In real life, if you had to download applications from untrusted source, what would you do to ensure the security of your device?
Answer:

- In real life, I would scan the application with my antivirus software as a preliminary check to make sure that it contained no obvious malware. Then, I would look online for any reviews and forums about that particular application to make sure that no one else had encountered any malware while using it.