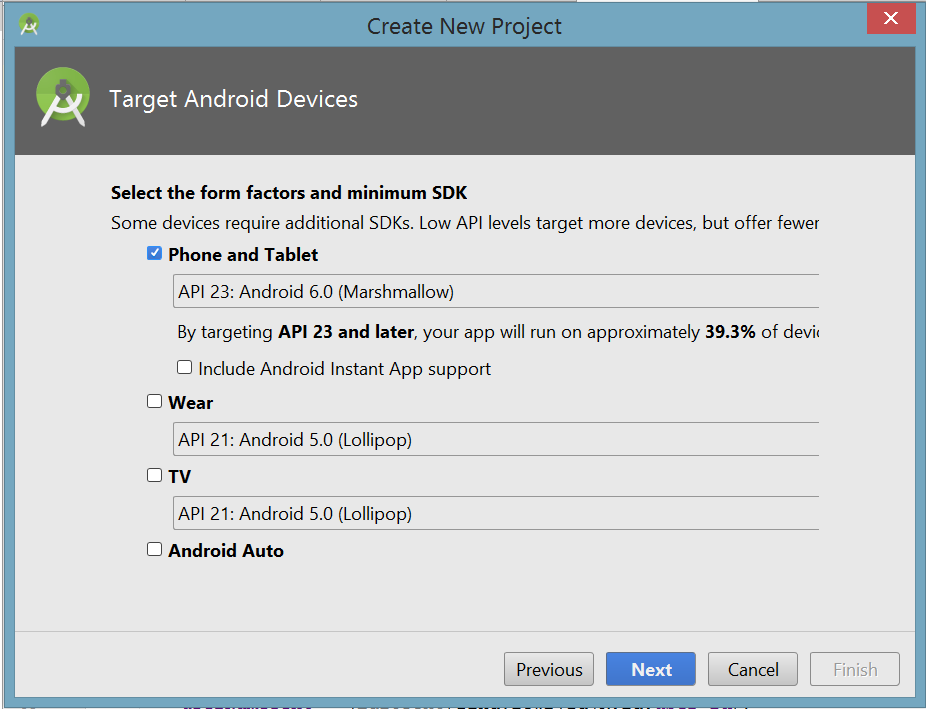
**Notes:**

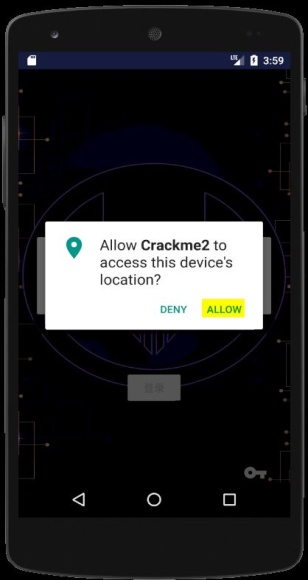
* this crackMe challenge can be solved both on android emulator and on android device
* Android Studio version: 3.0.1
* API 23



**Before we start:**

let's install the apk into the emulator and run the app.

we can see that that the app request to use a location permission which means we might need to be somewhere in order to login.

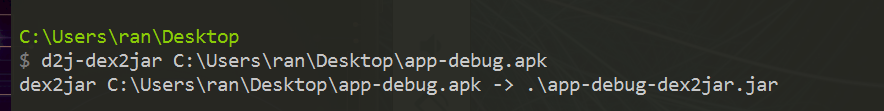


after we accept the request (or else continue to get this pop up) , we see another pop up in Chinese and we can't do anything in the application anymore.



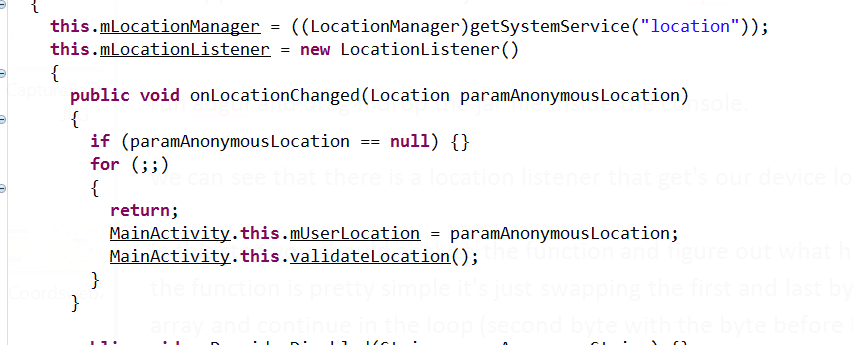
**Examine The Code (**[**go to solution**](#solution)**)**

run appie.exe and enter dex2jar command on the apk file.

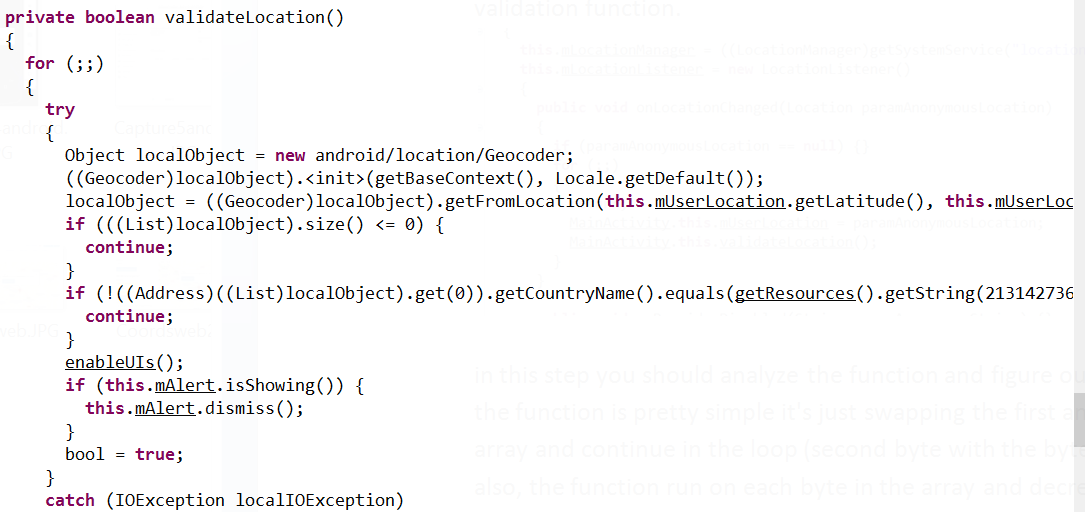
run adgui and drag n drop the jar file inside the console.



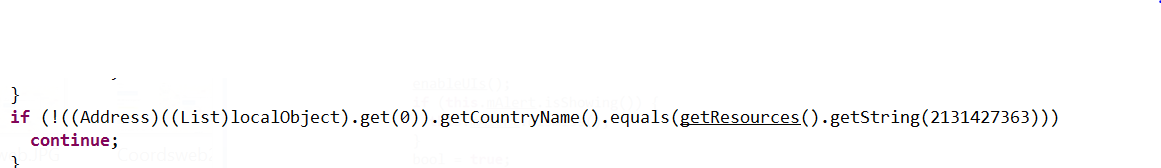
we can see that there is a location listener that get's our device location and call a validation function.



we can see that the location is used to get on which country we are at by Geocoder



we see here better that the country we are at is being check with some string in the resources.

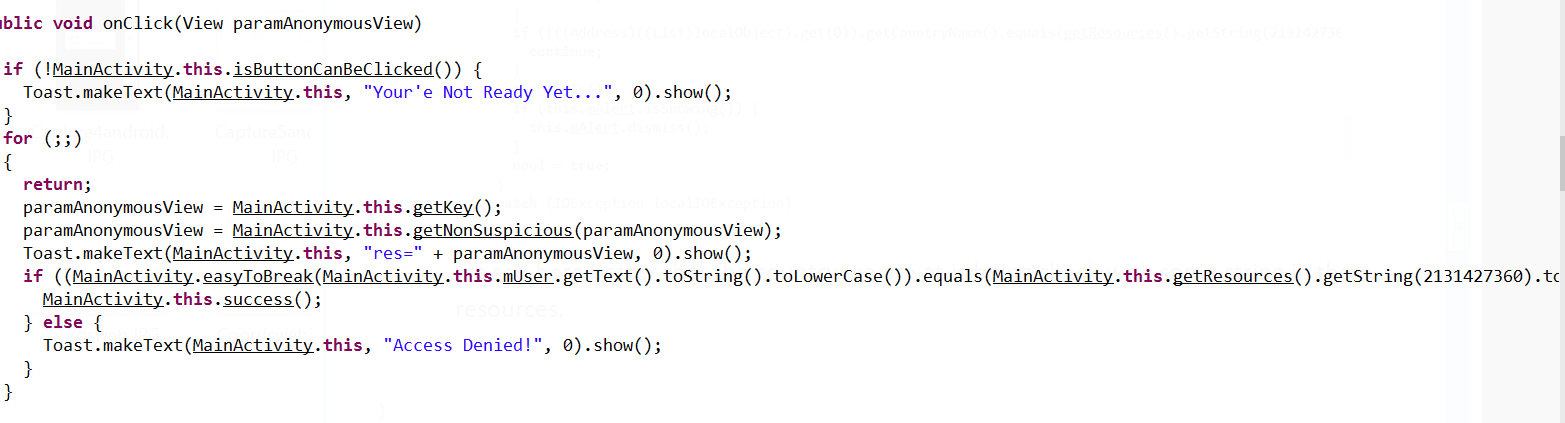


we can notice a lot of functions but it's hard to figure them out yet.

the most important function is the onclick listener of the login button.

*\*I removed the toast message so it will more clear that the button disabled.*

*.*



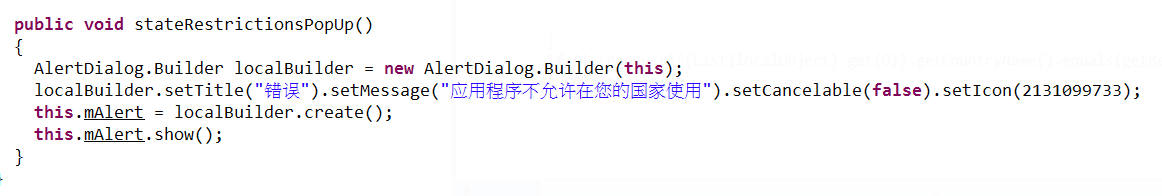
we can see that there are two validation functions ("easyToBreak" and "hardToBreak")





*\*hint - the name of the function actually means right*

the last and most important is the pop up that block our way.



**The Solution**

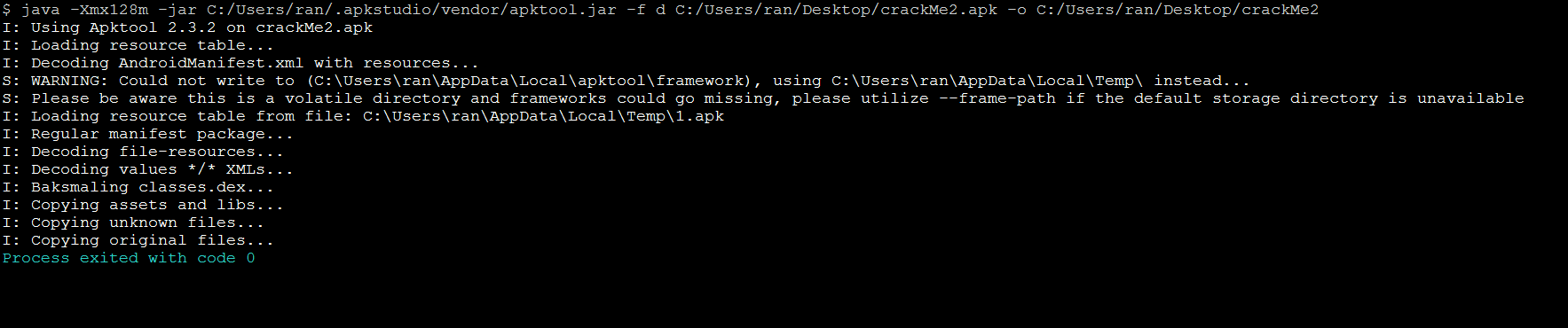
**Step 1:**

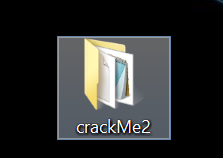
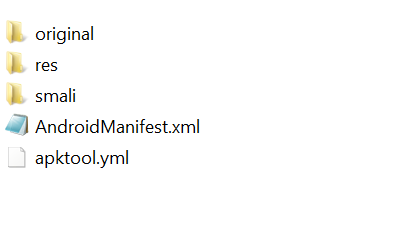
there are a lot of ways to solve this step but I'll show only two.

the first way is by understanding the all the app design and the Chinese letters (might use translator - https://www.freetranslation.com/en/translate-english-chinese) to understand that the app is only active in China.

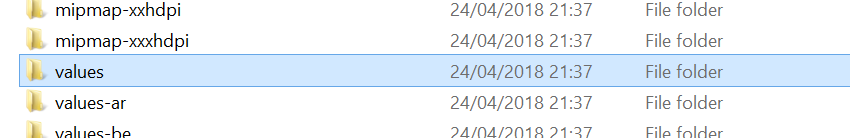
title= "error" , message= "The application does not allow you to use in your country"

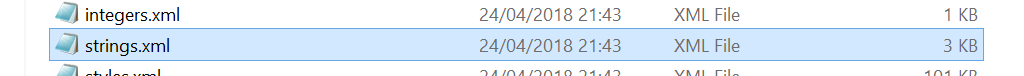
the second way is to run apktool on the jar file.

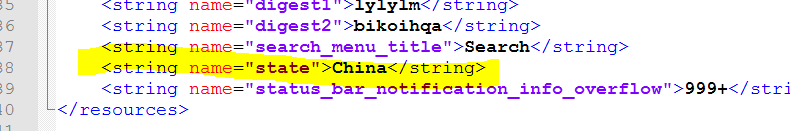




go to "res" -> "values" -> "strings.xml"



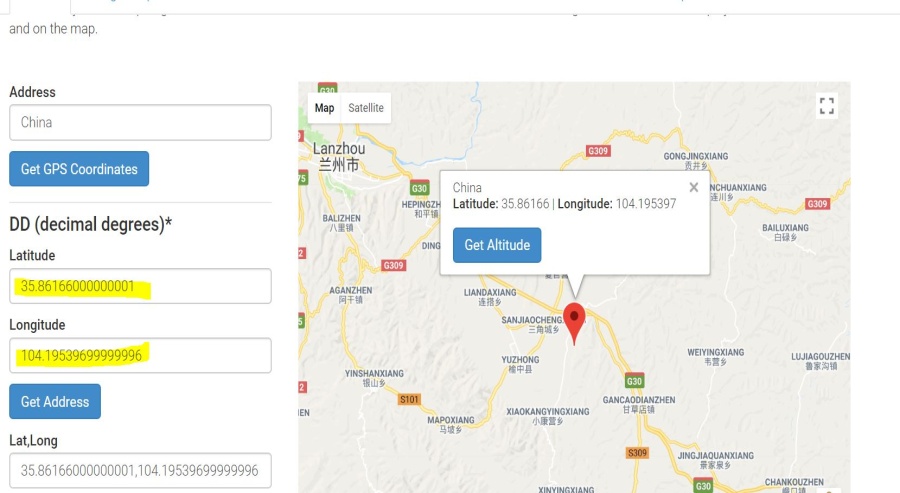




**Step 2:**

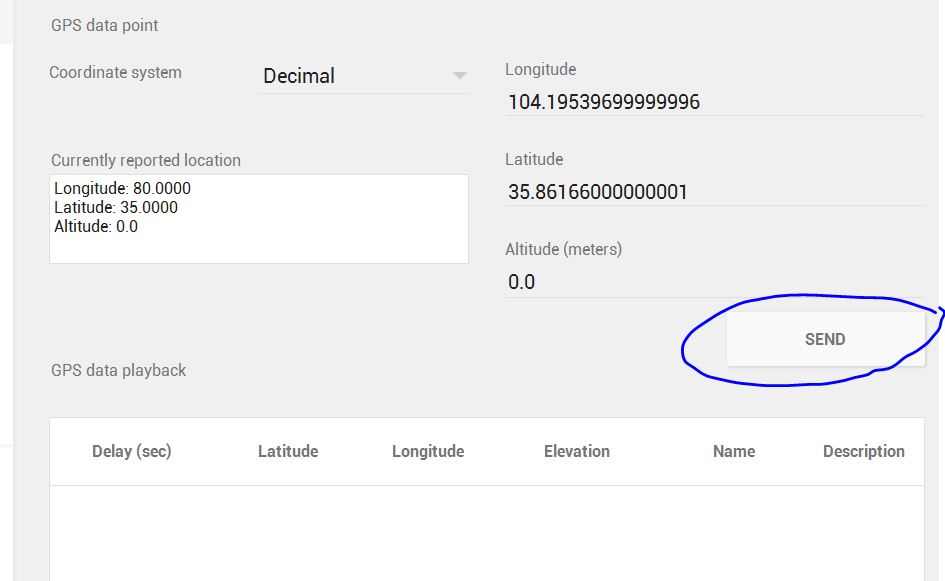
go to https://www.gps-coordinates.net or some other website to get coordinates.

get coordinates of a place in china and copy the latitude and longitude.



**Step 3:**

if we use emulator we can simply change the the latitude and longitude of the device. I used android studio emulator. (for Genymotion https://stackoverflow.com/questions/30709329/give-fake-location-to-genymotion-emulator)



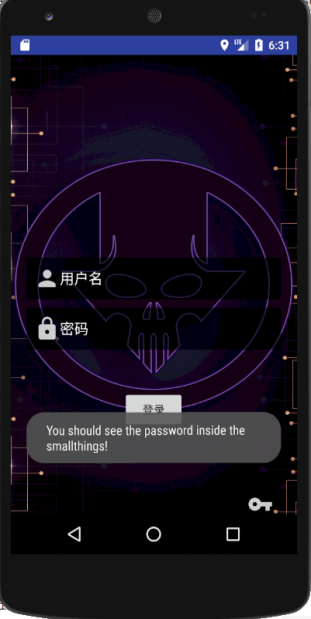
\*in real device you can download mock fake gps application or that can be done in several ways. (or you can go to China with the device it will work too :) )



https://play.google.com/store/apps/details?id=com.lexa.fakegps&hl=en\_US

now the app unlocks:

We can see that if we try to press the login button nothing happens (the button can't be clicked) and when we press the small key button at the bottom this message appears



**Enable the login button**

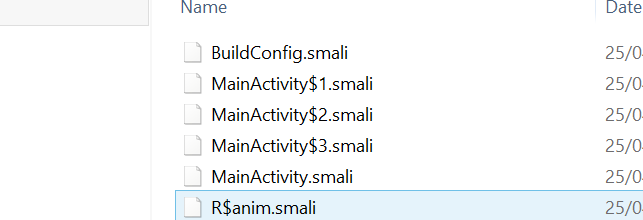
in the decompiled folder we'll go to the "smali" folder to check if we can make the login button to be clicked again, which means change the isButtonCanBeClicked()

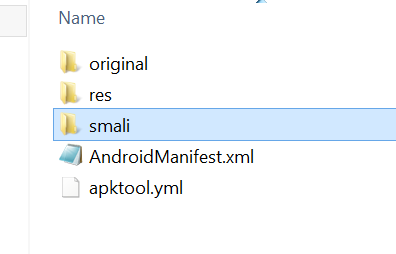
(that we saw in the jdgui console) to be true.

*\*there are lot more ways doing this phase ( I choose to change the condition that check if the button clicked to be the opposite).*

**Step 1:**

go to smali -> com -> example -> ran -> crackme2 ->

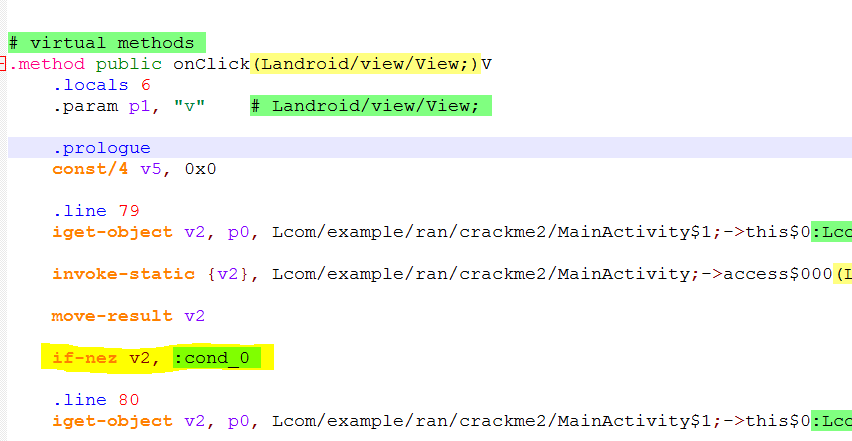




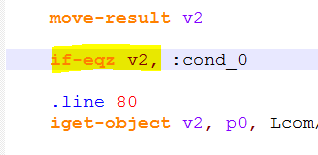
**Step 2:**

open MainActivity$1.smali file.

in line 54 we can see the condition - let's change it to the oposite.

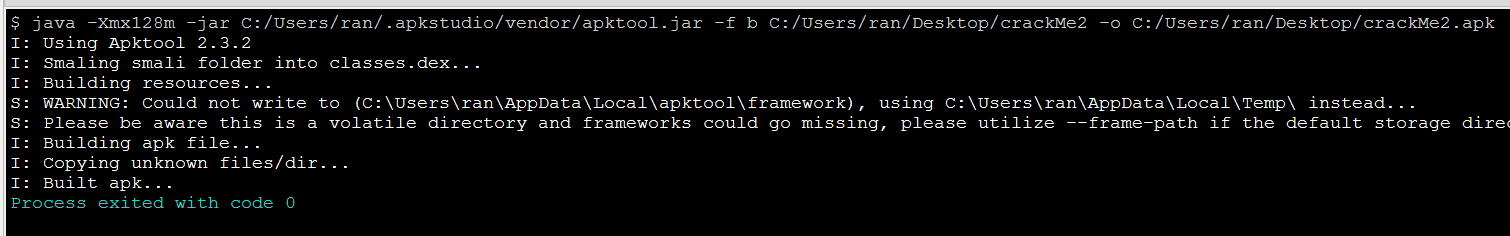


replace "if-nez" to "if-eqz" and save the file.



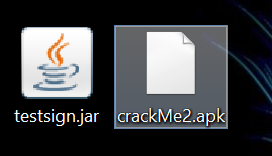
**Step 3:**

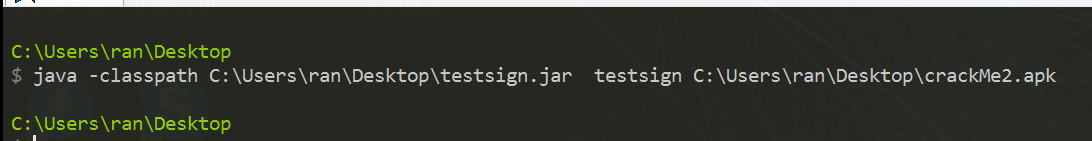
build the decompiled folder with apktool.



**Step 4:**

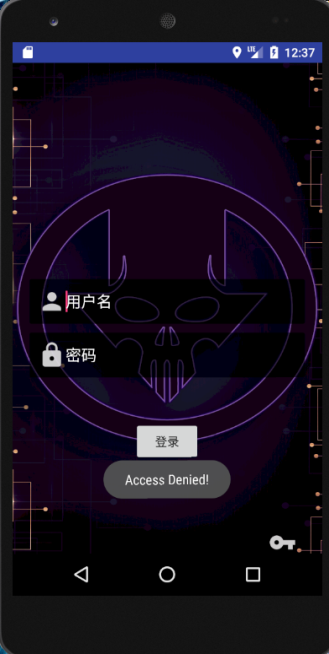
download testsign.jar and sign the apk.





**Step 5:**

uninstall the application from the emulator and install the new apk.



we can press the button now!

**Find The USER**

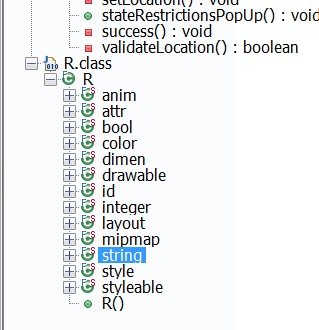
**Step 1:**

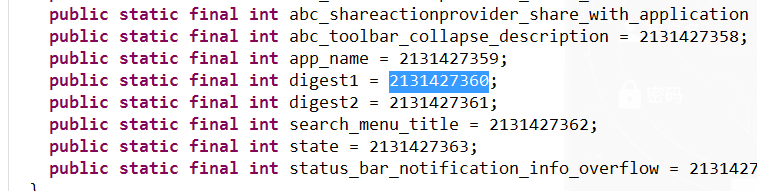
in the jdgui consol we saw that the user name is being calculated in the function "easyToBreak" with some string called "digest1" from the resources folder.

by the name of the function will examine the code and see if it's easy to break.

**Step 2:**

in the jdgui consol go to "R.class" -> "string" and press ctrl + f to find to number in the resources.

****

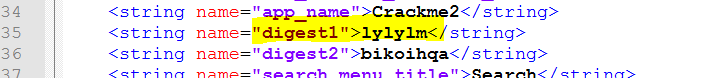
****

we can see that the number belongs to the variable "digest1"

**Step 3:**

lets search for "digest1" in the decompiled folder (from the apktool command from steps before).

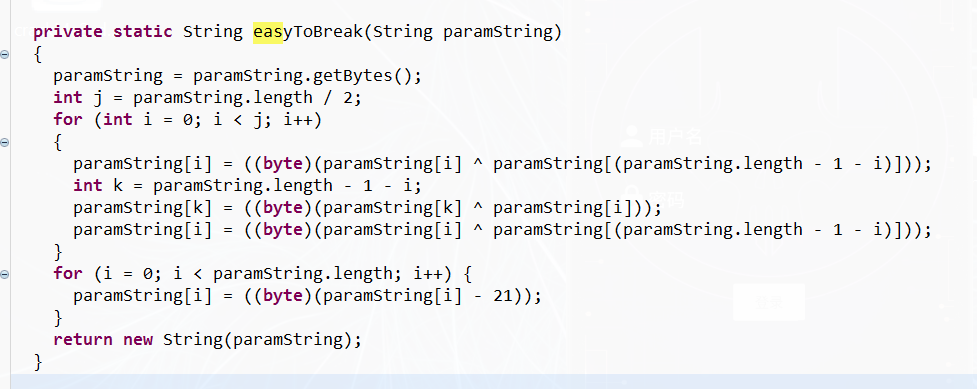
again will go to go to "res" -> "values" -> "strings.xml"

****

let's copy the string "lylylm".

**Step 4:**

now will check the function "easyToBreak" in jdgui

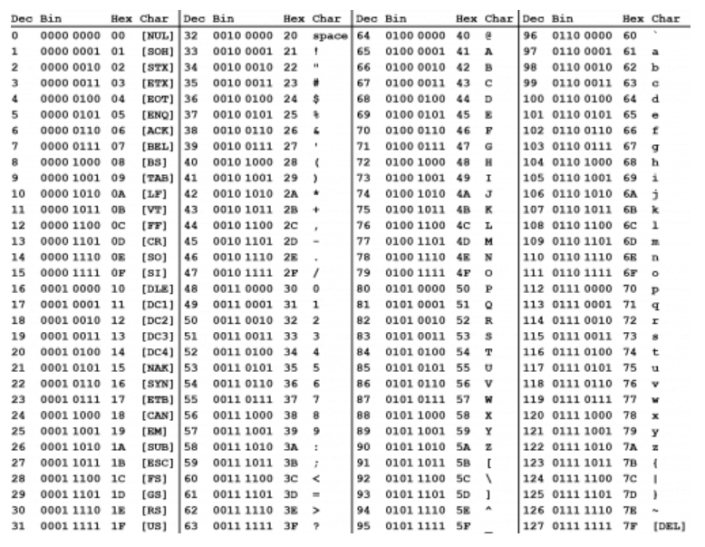
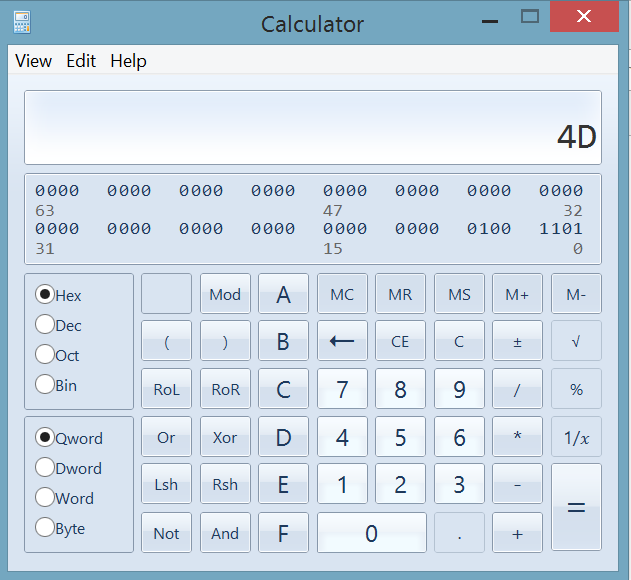
****

**Step 5:**

in this step you should analyze the function and figure out what how to reverse it.

the function is pretty simple it's just swapping the first and last byte in the bytes array and continue in the loop (second byte with the byte before last and etc.)

also, the function run on each byte in the array and decrease each byte(char) with 0x15. which means if the first character in the byte array was 'a' (hex value of 61) so after the decrement the char value will be 'L' (61 - 15 = 4c ). you can use hex in calculator in windows and an ascii table for this.

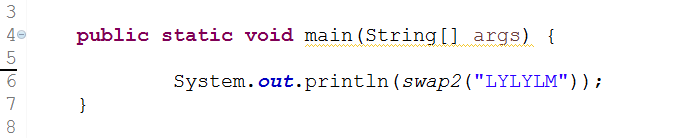


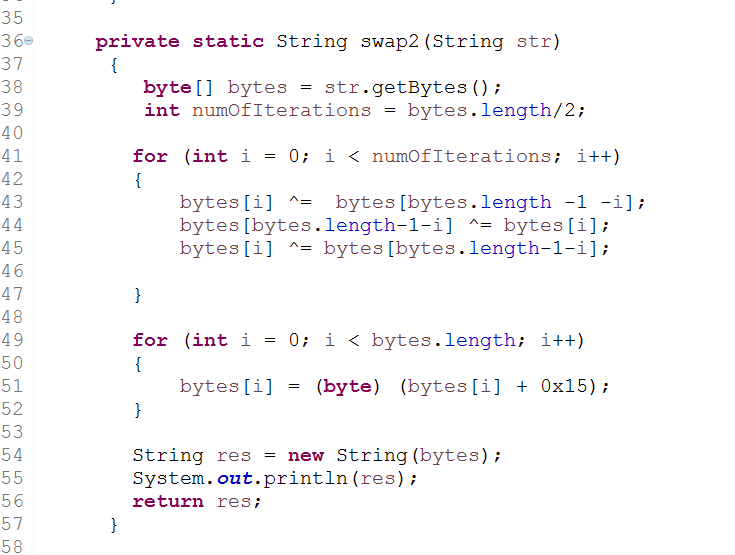
**Step 6:**

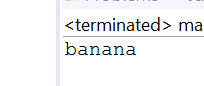
open online java compiler (https://www.tutorialspoint.com/compile\_java\_online.php) or java IDE like eclipse.

copy the function two times. the second time the function going to swap the bytes again so it will be sorted in the correct original order. the next thing we need to do is add the 0x15 to each byte. run the code with the string "LYLYLM" that we copied in **step 3** (notice that the resource string is **small caps** and in the function it's being **uppercased**).

print the result to get the user name = "banana".





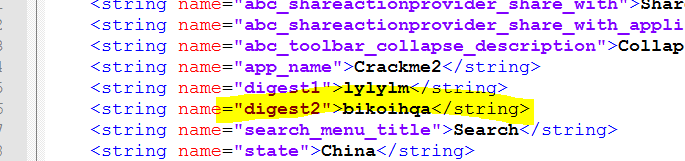


USERNAME = "banana".

**Find the PASSWORD**

**Step 1:**

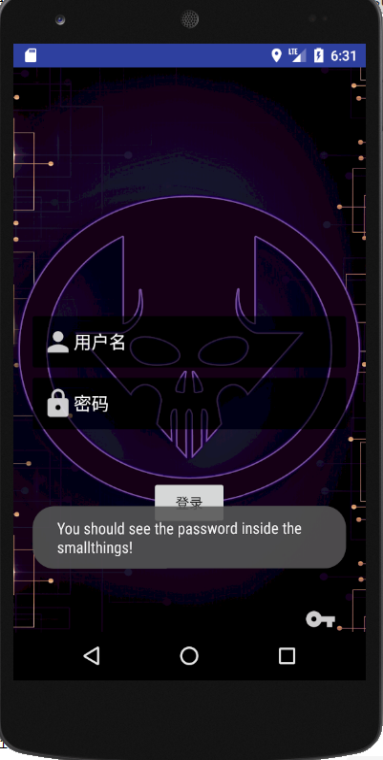
like the user name we can see that the password is calculated and being checked with the string "digest2" that is in the resources (same method we found the user name digest1)



digest2 = **bikoihqa**

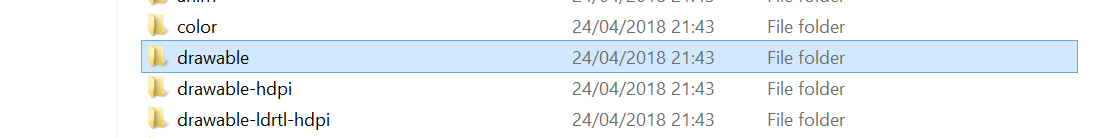
**Step 2:**

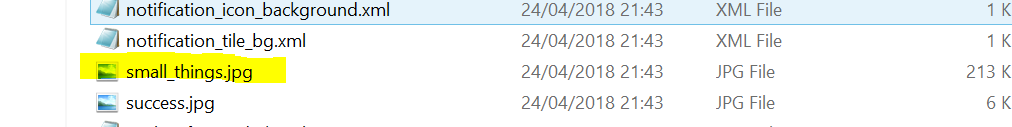
in jdgui we can see that there is no easy way to crack the "hardToBreak" function.

let's see what happened when we pressed the small key icon in the app.

there is a toast message saying that **the password is inside the "smallthings"**.

after we dig the application files little bit we see that there is an image file called "smallthings" (or we can search for the text "smallthings" and find the file in drawable folder.





**Step 3:**

when we open the file we get this image

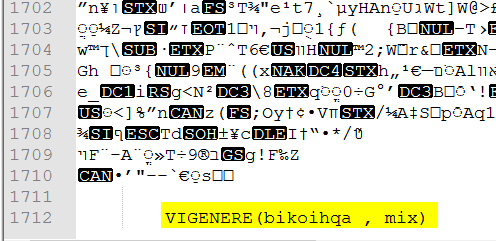


if we pay attention to the message it said the password is inside the file (another way to think about it).

let's open the image in notepad and see if something is hidden there.

**Step 4:**

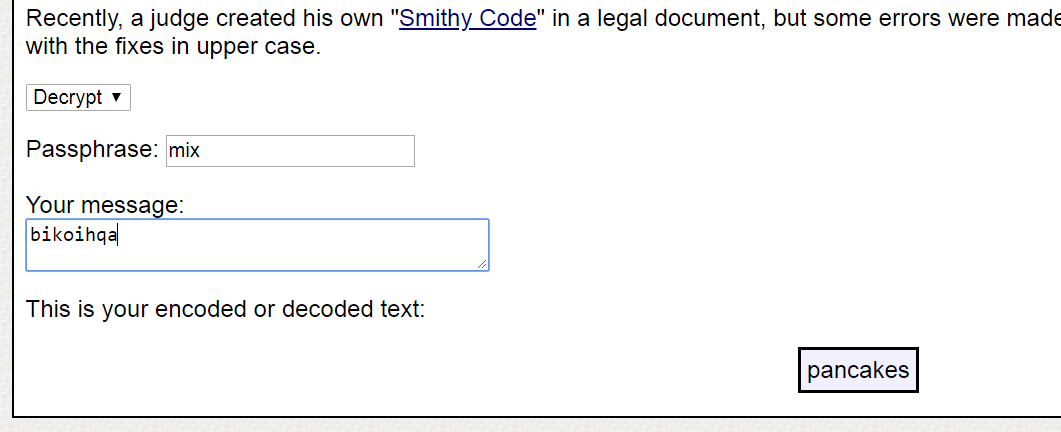
after we opened it in notepad we see this hidden text



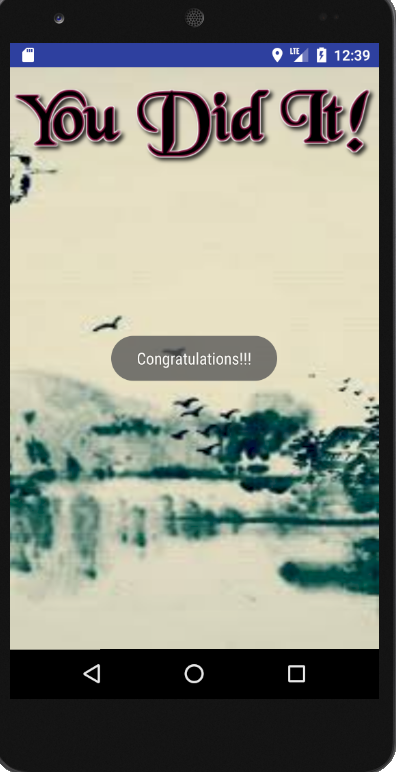
**Step 5:**

let's go to rumkin.com to decode the text "bikoihqa" (which is the "digest1") with the key "mix" and see what will get.

http://rumkin.com/tools/cipher/vigenere.php



PASSWORD = "pancakes"





\*\*\*end of challenge\*\*\*