Name: Ghulam Abbas
Student ID: Fa22/BSDFCS/030
Section: A
Assignment submitted to: -
Ms. Fatima

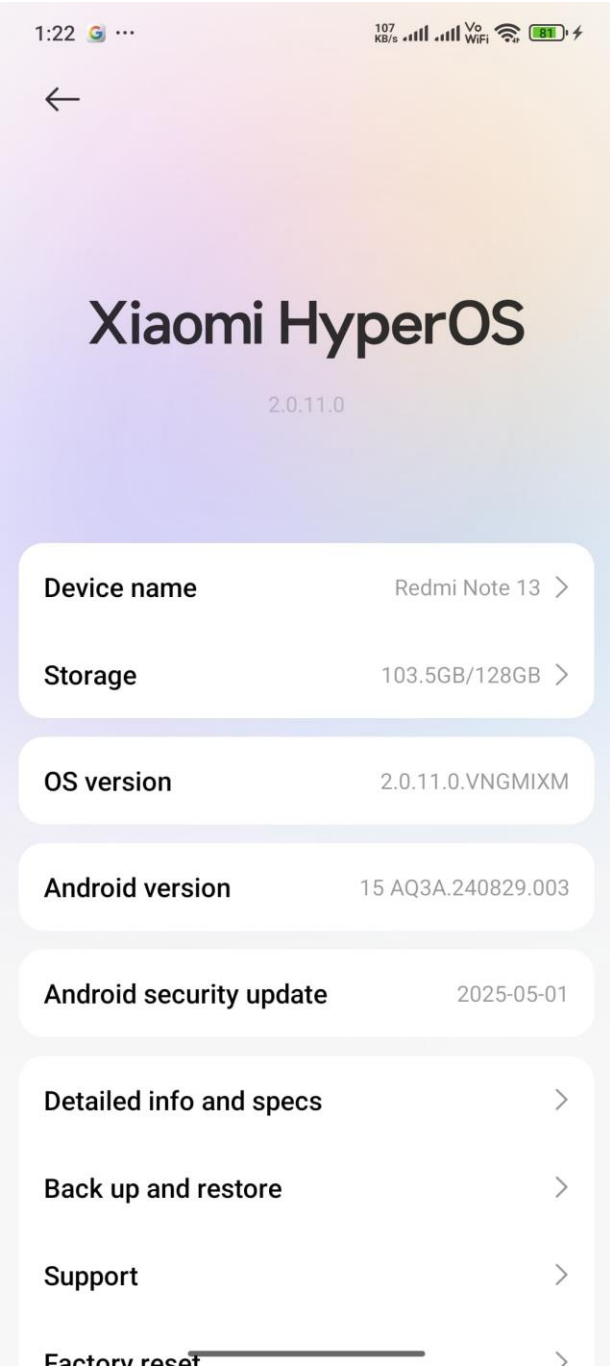
Android Logical Data Acquisition Using ADB Tools

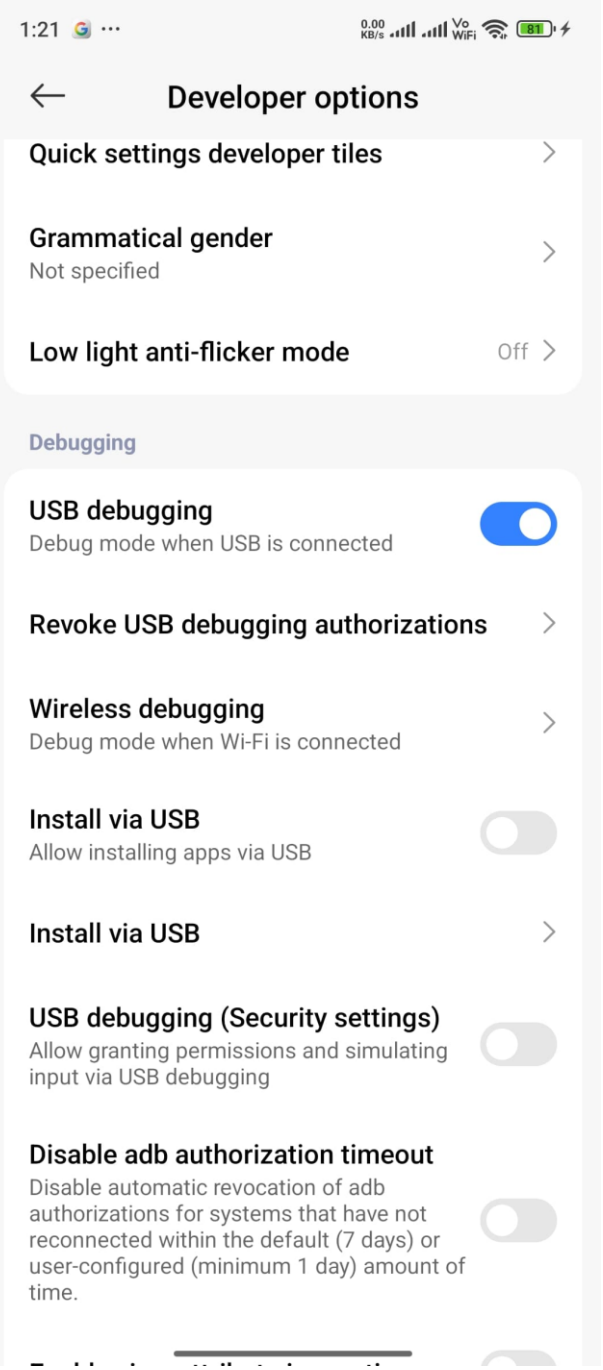
Step 1: Install tools that are below mentioned with links for logical data Acquisition

- ADB Tools (https://developer.android.com/studio/releases/platform-tools)
- Java (https://javadl.oracle.com/webapps/download/AutoDL?BundleId=252044_8a1589aa0fe24566b4337beee47c2d29)
- Android Drivers
- Android backup extractor master (https://github.com/nelenkov/android-backupextractor/releases/download/latest/abe-62310d4.jar)
- Hashmyfile (https://github.com/forenipackages/hashmyfiles/blob/master/HashMyFiles.exe)

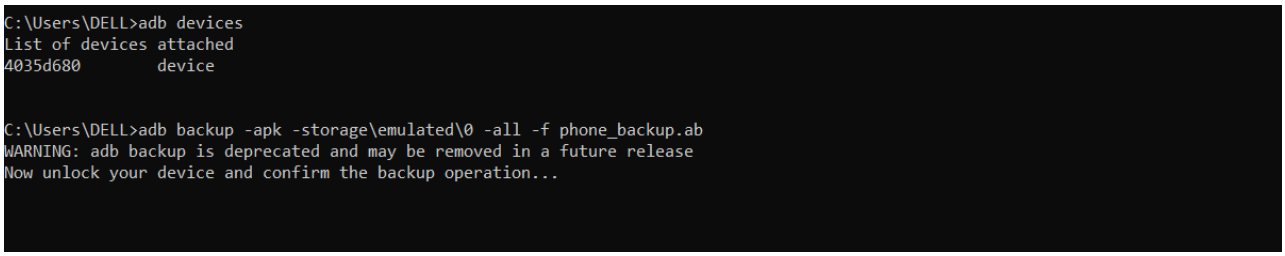
Step 2: Turn on USB Debugging through Developer Options on your Android device. If you don't see Developer Options in the settings, navigate to Settings > About phone, then tap the Build Number (or OS version) seven times to unlock it.

MY DEVICE NAME **REDMI NOTE 13** (MODEL **23129RAA4G/ android15**)

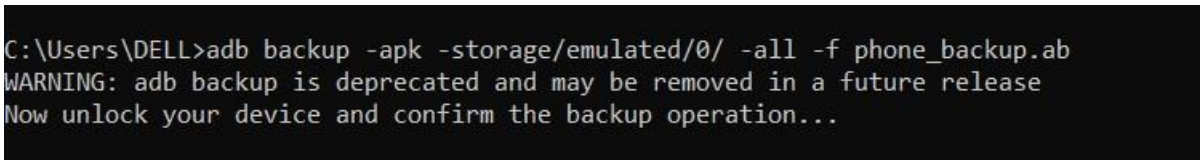




Step3: After on the debugging option on android phone connect the android device with workstation through USB. After that run command and ***adb devices*** in workstation command prompt to check device are given response or not and then run command for backup



adb backup -apk -storage\emulated\0 -all -f phone_backup.ab

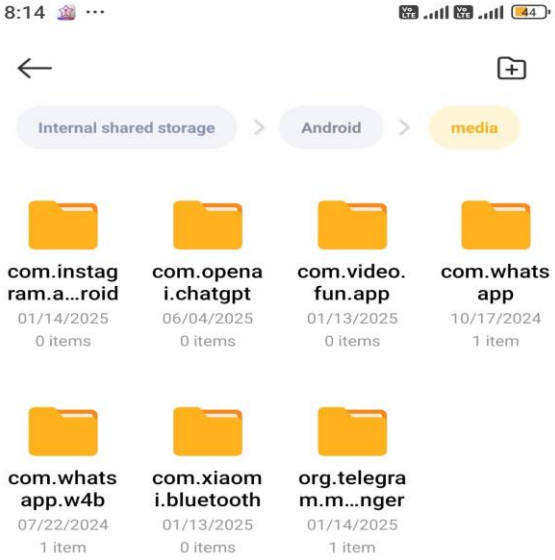


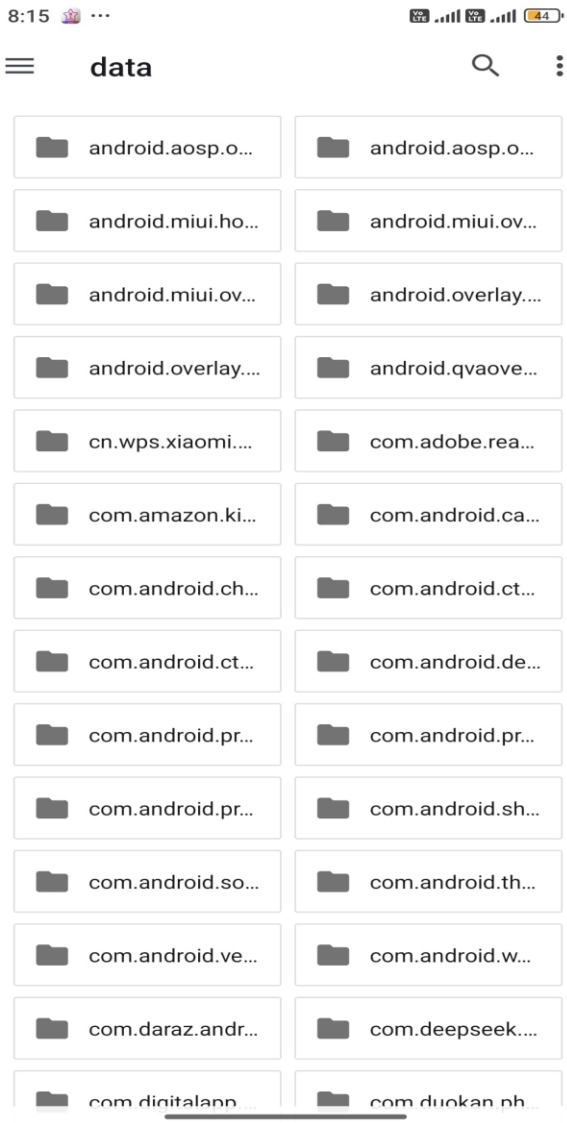
After that run command ***adb backup -apk -storage\emulated\0 -all -f phone_backup.ab*** to take the backup of the apps and shared folders like media on the workstation. Follow the onscreen instructions provided by adb tool.

What this path (storage\emulated\0)gives?

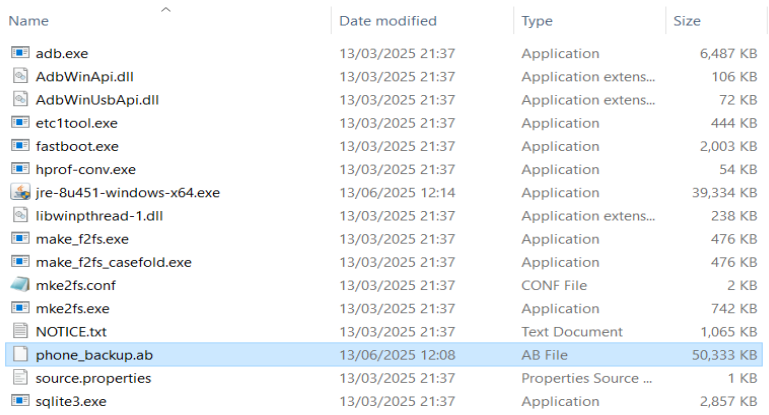
The path /storage/emulated/0/Android/media/ shown in the screenshot gives access to app-specific media files stored on internal shared storage. It contains folders like com.whatsapp, org.telegram.messenger, and com.instagram.android, which include user-visible data such as images, videos, voice notes, and downloaded content from these apps. This path does not contain sensitive internal app data like databases or login info, but it is

useful for retrieving media files without needing root access.





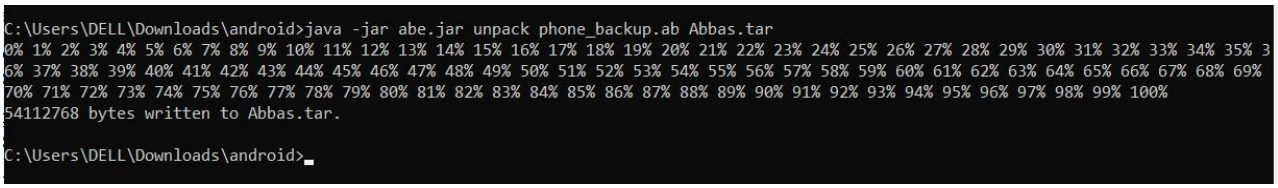
As you can see backup file phone_backup.db is completed and stored on workstation



Name	Date modified	Type	Size
adb.exe	13/03/2025 21:37	Application	6,487 KB
AdbWinApi.dll	13/03/2025 21:37	Application extens...	106 KB
AdbWinUsbApi.dll	13/03/2025 21:37	Application extens...	72 KB
etc1tool.exe	13/03/2025 21:37	Application	444 KB
fastboot.exe	13/03/2025 21:37	Application	2,003 KB
hprof-conv.exe	13/03/2025 21:37	Application	54 KB
jre-8u451-windows-x64.exe	13/06/2025 12:14	Application	39,334 KB
libwinpthread-1.dll	13/03/2025 21:37	Application extens...	238 KB
make_f2fs.exe	13/03/2025 21:37	Application	476 KB
make_f2fs_casefold.exe	13/03/2025 21:37	Application	476 KB
mke2fs.conf	13/03/2025 21:37	CONF File	2 KB
mke2fs.exe	13/03/2025 21:37	Application	742 KB
NOTICE.txt	13/03/2025 21:37	Text Document	1,065 KB
phone_backup.ab	13/06/2025 12:08	AB File	50,333 KB
source.properties	13/03/2025 21:37	Properties Source ...	1 KB
sqlite3.exe	13/03/2025 21:37	Application	2,857 KB

Step 4: This screenshot shows the successful unpacking of an Android backup file (.ab) into a standard tar archive (.tar) using the Android Backup Extractor (abe.jar) tool.

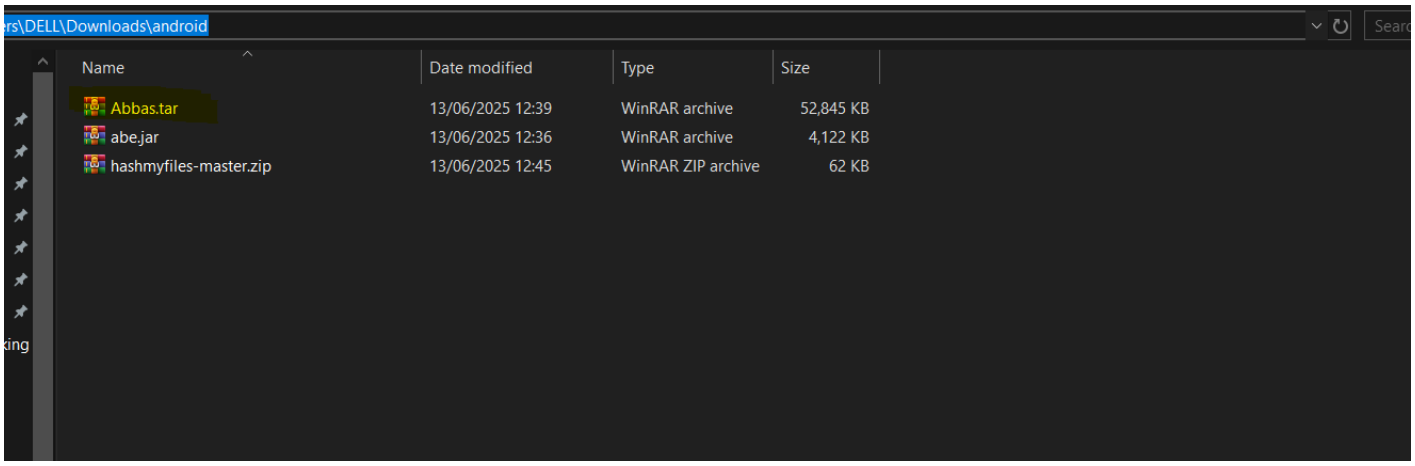
The command java -jar abe.jar unpack phone_backup.ab Abbas.tar is used to extract an Android backup file (.ab) into a standard .tar archive using the Android Backup Extractor tool. In this process, the abe.jar file (a Java program) reads the backup file phone_backup.ab and unpacks its contents into Abbas.tar, making the data accessible for further inspection. The screenshot shows the unpacking progress from 1% to 100%, and finally confirms that 54,112,768 bytes (approximately 54 MB) were successfully written to the Abbas.tar file. This tar file can now be opened with tools like 7-Zip or WinRAR to view the extracted contents.



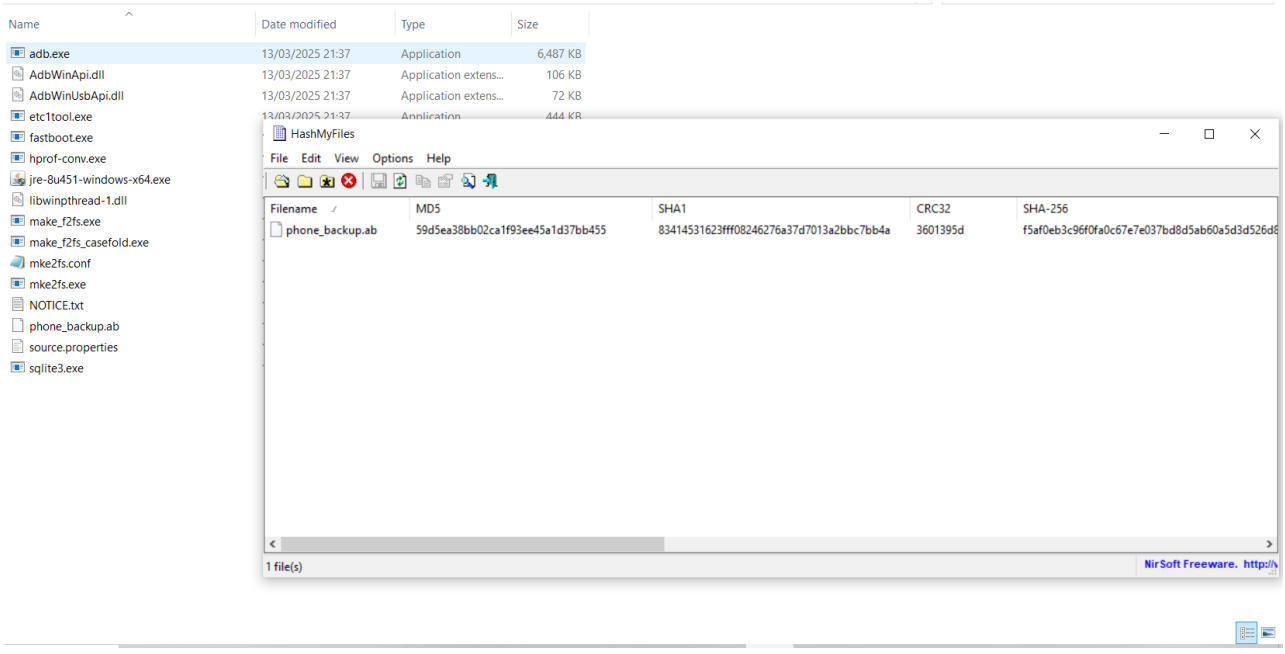
```
C:\Users\DELL\Downloads\android>java -jar abe.jar unpack phone_backup.ab Abbas.tar
0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 24% 25% 26% 27% 28% 29% 30% 31% 32% 33% 34% 35% 3
6% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 56% 57% 58% 59% 60% 61% 62% 63% 64% 65% 66% 67% 68% 69%
70% 71% 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 84% 85% 86% 87% 88% 89% 90% 91% 92% 93% 94% 95% 96% 97% 98% 99% 100%
54112768 bytes written to Abbas.tar.

C:\Users\DELL\Downloads\android>
```

Converted into .tar file and stored on a workstation



Step 5: After converting take hashes using command *certutil -hashfile Abbas.tar SHA1* if on Windows and *sha251sum phone_backup.ab* if on Linux/Mac to maintain integrity of the evidence. I'm using hashmyfile for this purpose on windows



Generated hashes successfully

Md5: 59d5ea38bb02ca1f93ee45a1d37bb455

SHA1: 83414531623fff08246276a37d7013a2bbc7bb4a