Assignment: KEYS, ADDRESSES, AND WALLETS

Q1. 1. Difference between hardened and non-hardened keys

Hardened keys add an extra layer of security when generating child keys from a parent key.
In simple terms, a **hardened key** can only be derived using the parent's *private key*, while a **non-hardened key** can be derived using just the *public key*.

So if someone gets access to a non-hardened child private key and its parent public key, they can trace back to other keys which is a security risk.
That's why hardened keys are safer, especially when dealing with sensitive wallets or funds.

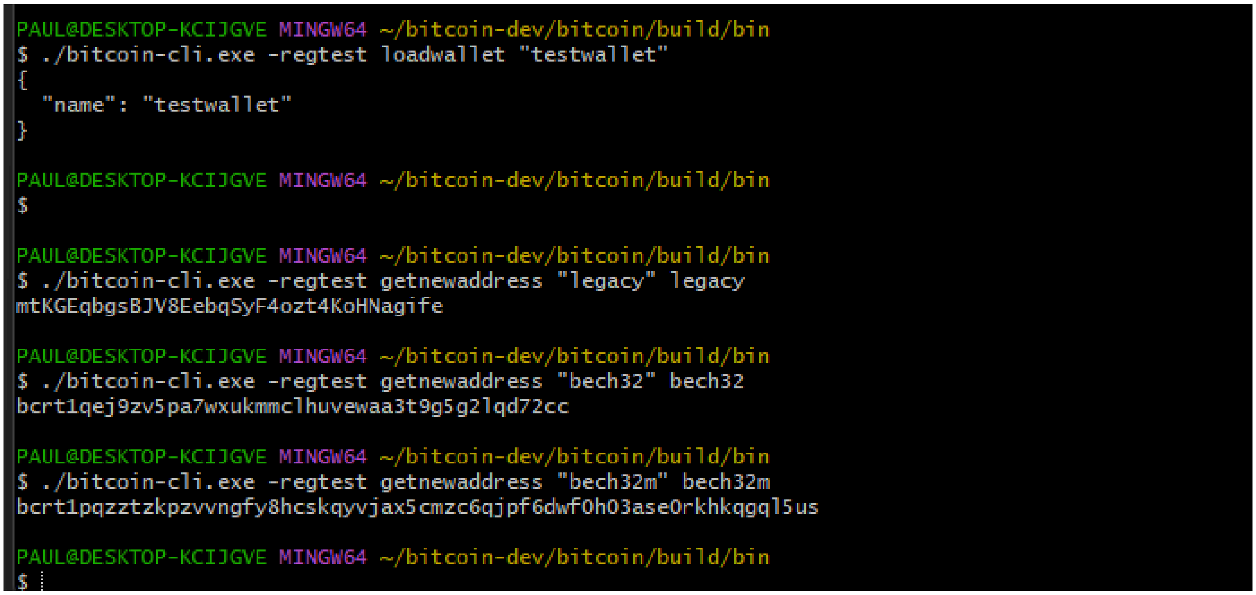
Q2. 2. Why wallet developers prefer deterministic wallets over non-deterministic wallets

Developers prefer **deterministic wallets** because they can generate all future addresses from one master seed phrase (usually 12 or 24 words).
This makes **backup and recovery very easy** if you lose your wallet file, you just need the seed to restore everything.

Non-deterministic wallets, on the other hand, generate random, unrelated keys meaning if you don't back each one up individually, you could lose access to some funds.

So in short: deterministic wallets are simpler, more organized, and safer for users.

Q3.



```
PAUL@DESKTOP-KCIJGVE MINGW64 ~/bitcoin-dev/bitcoin/build/bin
$ ./bitcoin-cli.exe -regtest loadwallet "testwallet"
{
  "name": "testwallet"
}

PAUL@DESKTOP-KCIJGVE MINGW64 ~/bitcoin-dev/bitcoin/build/bin
$

PAUL@DESKTOP-KCIJGVE MINGW64 ~/bitcoin-dev/bitcoin/build/bin
$ ./bitcoin-cli.exe -regtest getnewaddress "legacy" legacy
mtKGEqbgsBJV8EebqSyF4ozt4KoHNagife

PAUL@DESKTOP-KCIJGVE MINGW64 ~/bitcoin-dev/bitcoin/build/bin
$ ./bitcoin-cli.exe -regtest getnewaddress "bech32" bech32
bcrt1qej9zv5pa7wxukmmclhuvewaa3t9g5g2lqd72cc

PAUL@DESKTOP-KCIJGVE MINGW64 ~/bitcoin-dev/bitcoin/build/bin
$ ./bitcoin-cli.exe -regtest getnewaddress "bech32m" bech32m
bcrt1pqzztzkpzvvngfy8hcskqyvjax5cmzc6qjpf6dwf0h03ase0rkhkqgql5us

PAUL@DESKTOP-KCIJGVE MINGW64 ~/bitcoin-dev/bitcoin/build/bin
$
```