Question 1: Which transaction history will be kept as the canonical version? (Specify the sequence using the italicized lowercase letters beneath each block and indicate the criterion you used to make your decision.)

The rule I use here is the “longest chain” rule, which is used in many blockchains like Bitcoin, Ethereum, etc.
Through the “longest chain” rule, at this point in time, we can have many answers to the question.
b-->e-->f-->g-->h
c-->e-->f-->g-->h
a-->d-->f-->g-->h
These are the answers, and none of them is prioritised than others. Therefore, we need to wait until one of them becomes the longest to choose that one.

Question 2: From the two questions below, please answer the one that is most appropriate given your response to Question 1.

- ***If you are not able to identify the transaction history, explain why, and then describe how the network will eventually arrive at a canonical version.***
- ***If you are able to identify the transaction history using one of the two standard criteria, list the blocks that will become orphans.***

I was not able to identify the transaction history. I described the case in the last question. Due to the “longest chain” rule, all of them have the opportunity to become the longest very soon. So, we have to wait until one of them becomes.
As soon as one of these chains become the longest, meaning that it has more blocks on its chain than the others, the chain is broadcast to the network and all nodes agree upon it.

Question 3: As you have learned, it is perfectly normal for multiple versions of the blockchain data structure to exist at one time (but not indefinitely). Over time, the network "cleanses itself" by eliminating orphan blocks and settling onto a single version of the transaction history. If there is no data tampering, why are the block header hash values different for blocks a, b, and c in the above diagram?

There can be many reasons:

1. Block *a* differs from the two others because it has a different successor (block *d*) than the others. Even if block *a* has the exact same transactions, because it points to another block, the hash of the block changes.
2. Block *a* has a different difficulty. This means even if it pointed to the same successor, and had the exact same transactions, the *Nonce* was different, therefore a different hash.
3. Block *b* has a different block creator than block *c*, since the name of the block creator, i.e., block submitter, comes in the block header, even if both blocks *b* and *c* have the same transactions, they have different hashes.