A-4 What is an anonymity set and why is it important that it is large? Given two anonymity sets A_i and A_j, $(i \neq j)$, how would you interpret $AS_i \cap AS_j$?

Answer: "The set of people in which you are anonymous". The larger this set is, the more anonymous you will be. If you are part of a set of only 2 participants, there's a 50% chance to for an adversary to "guess" right, since you must be one of the two. However, if the set has 100 participants, the chance is only 1 out of a 100, 1%. The intersection between two sets such as those can be used as an attack vector.

A-5 What is the purpose of the random value R_0 in a Mix?

Answer: Avoiding guessing attacks by introducing randomness in the message. If R_0 was not used, the output from the mix would be $K_a(R_0, M), A$ which would be possible to verify by simply encrypting with the public key (which is available to everyone).

A-6 What is the purpose of the random values R1 in a Mix?

Answer: The randomness R_1 hides the input-output correspondence. A message without R_1 would be $K_1(K_a(R_0, M), A)$. If you take the output, $K_a(R_0, M), A$ and encrypt with the public key of the mix you could easily find the corresponding input message.

A-9 Describe the strongest adversary possible and explain why two fundamentally different anonymizing designs have intrinsically different possibilities of protecting against such an adversary.

Answer: The Global Passive Adversary, GPA. It has the ability to observe each and every node in a network. It can see every in-out traffic for every node. Thus, it knows the source as well as destination addresses for every packet in the network.

High-latency design: tries to protect against GPA. In this design, it is assumed that others can know the source and destination addresses of any given packet in a network. A mix is used for this. The mix hides the correspondence between input-output. By having the design operating in a high-latency way (collect messages over a time period, before forwarding a large number of them out periodically), there is a time delay between the input-output which hinders the GPA from seeing corresponcenses. Having the mix behaving this way increases the anonymity set.

Low-latency design: cannot defend itself against GPA. Low-latency means that there is little delay between input-output. The anonymity set for a lone mixer using low latency would be rather small. A GPA can easily use timing to trace the path of a message, and quite easily find the sender-reciever connection. What a low latency network can do is making it somewhat harder for a GPA, using several mixes, dummy traffic, mix the order of packets et cetera. But in the end, low-latency designs are theoretically not able to protect themselves from a GPA.

A-22 If you are using the Tor network for you own communication, would you be more or less safe if you would participate as a relay for others in the network as well?
Answer: You would be more anonymous doing this. By virtue of being a relay for others, you will generate traffic that is not only your own. The packets are encrypted, and thus a potential adversary can't know if cells leaving the node are generated here or relayed from another cell.

A-23 Several users can use the same exit node in Tor, but different intermediate nodes. How can the exit node know where to send the response from the target?

Answer: The circuit is open for 10 minutes, so it simply sends back to on the circuit with the corresponding Stream ID. The (immediate) node it sends to is the one with the correct Circuit ID.

A-24 Alice is negotiating keys during a chain construction in Tor. It is reasonable to assume that sending material to and back again from OR1 takes some time. Can she use this time to prepare for negotiating with OR2, OR3, ...? How/why not?

Answer: She could choose her x_i and calculate $g^{x_i} mod g$, but not encrypt it since the shared key, $K_1 = (g^{x_1})^{y_1} mod p$, which is a important part in making Alice anonymous, is created in the setup between Alice and OR_1.

A-26 A TCP handshake consists of the client and the server exchanging three messages: SYN, SYN-ACK and ACK. Explain why, in Tor, Alice can connect to a webserver and expect the TCP handshake with the server to be performed with low latency?

Answer: AES-128 bit, countermode, is used. This symmetric encryption scheme is quite fast, so the time it takes to encrypt for every step becomes quite low. The Onion Router n, OR_n, which in this case performs the handshake with a website, decrypts what it recieves from $OR_n - 1$. By inspecting the recognized and digest fields it determines that it is the final node. Now, the handshake is performed. The TCP handshake itself is performed only between the exit node and server itself. The delay here is negligible.