**Computer Systems 2022/23 Exam – Feedback/Possible Solutions**

**Question 1.**

**(a)** The data type cannot be floating point (*float* or *double*) because this would not overflow, but instead go to infinity. Therefore, it must be *short*, *int* or *long*; in order to find out which, we consider at what point the overflow occurs.

The maximum value for *int* is $2^{31}$-1, which is larger than $20^7$ and smaller than $20^8$; therefore, for *int*, the results go wrong when n >= 8. Therefore the data type is *int*.

**(b)** In the first technique, call to subroutine stores the return address in memory at the location of subroutine and then jumps to the instruction following that in memory. The return instruction will load the PC with the return address stored at the start of subroutine.

In the second technique, the call to subroutine pushes the return address onto a stack and jumps to the subroutine. The return instruction pops the return address from the stack and loads it into the PC.

An disadvantage of the first style is that it cannot deal with recursion, while an advantage of the second style is that it can.

**(c)** The first outer loop executes n/2 times, which is O(n). The inner loop executes log(n) times, which is O(log(n)). The second outer loop executes n/4 times, which is O(n). Therefore the overall complexity = O(n * log(n)) + O(n) = O(n * log(n)).

**(d) This question would not be examined this year.**

**Question 2.**

**(a)** A deadlock is a problem that occurs in concurrency when processes wait forever for each other to free resources. For example, three process A, B and C could become deadlocked if A is waiting for B to free a resource, while B is waiting for C to free a resource, while C is waiting for A to free a resource.

**(b)** The minimum execution time is 0.2 * 800 + ((0.8 * 800) / 10) = 224 seconds. **(4 marks)** There are several potential reasons why you would be unlikely to achieve this in practice: for example, memory bandwidth, cache capacity, dependencies between apparently independent pieces of code, etc. The remaining **(3 marks)** would be given for showing a good understand of this.

**(c)** **(i)** The technical reason is because race conditions can occur. A race condition is where two threads could be updating the same value at the same time, meaning one of the updates is overwritten. As an example, if there are '2' pairs of boots left in stock, and two people buy it at the same time, then their threads could both see '2' as the quantity in stage (i) of the threads at the same time, and then

both calculate that '1' should be the new value in stage (ii) of the threads at the same time, then both update the stock to '1' in stage (iii) of the threads at the same time; despite the true quantity being '0'.

**(ii)** The program needs to identify the critical section (checking and updating the stock) and ensure only one thread and be in this section. For full marks, the answer will refer to locks or other mechanisms to ensure this.

**(iii)** The performance will have got worst because of the extra cost of protecting the critical section; now, only one thread can enter that section of the code at a time, meaning that those operations must be performed seqeuentially rather than concurrently.

**Question 3.**

**(a)** The answer should demonstrate understanding of one advantage **(1 mark)** and one disadvantage **(1 mark)** of public key crypto compared to symmetric key crypto; e.g. public key crypto has more security (due to lack of key exchange/key agreement beforehand) but is slower/more inefficient (with larger messages).

Then, using these differences, the answer should state **(1 mark)** and justify **(1 mark)** a preference for one of the two systems for use in this specific communication between Alice and Bob. Note that it does not matter which you state, as long as you justify it for the purpose of sending a message.

**(b)** Sockets are the interface between the Application and Transport Layers; a process sends and receives messages to and from its socket. Socket addresses are made up of an IP address and a port number. Bob's IP address uniquely identifies Bob's host machine, and the port number of the messaging process uniquely identifies that process. Therefore, Bob's socket address used in this communication uniquely identifies the messaging process running on his host machine.

**(c)** Before the handshake, both Alice and Bob's hosts are in LISTEN state.

In the first step of the handshake, Alice sends a SYN message to Bob. This message's TCP header has a sequence number x and SYNbit = 1.

After this message is sent, Alice's host is in SYNSENT state. Once it is received by Bob, Bob's host is in SYNRCVD status.

In the second step of the handshake, Bob sends a SYNACK message to Alice. This message's TCP header has sequence number y, acknowledgement number x+1, SYNbit = 1 and ACKbit = 1.

In the third step of the handshake, Alice sends an ACK message back to Bob. This message's TCP header has sequence number x+1, acknowledgement number

y+1 and ACKbit = 1.

Once Alice has sent this message, her host is in ESTAB state. Once Bob receives it, he is also in ESTAB state, and the connection is established.

**(d)** Demonstrate understanding of the purpose of the network layer in general **(3 marks)**; reference to this specific communication between Alice and Bob must be present in the answer to earn the other **2 marks**. Points may include:

- Every host/router has network layer protocols
- Has two key functions: forwarding and routing
- Forwarding means moving packets from router's input to appropriate router output
- Routing means determining route taken by packets from source/Alice to destination/Bob
- On sending/Alice side, it encapsulates segments into datagrams/packets
- On receiving/Bob side, it delivers segments to transport layer
- Third important function (in some architectures) is connection setup; i.e. two end hosts establish virtual connection