Very Short Answer:

(1) (2) What does GCD stand for?

(2) (2) Issuing many instructions per cycle (say, 12) puts tremendous pressure on what two parts of the machine?

(3) (2) Predicting the direction of a branch is not enough. What else is necessary?

(4) (2) What is the name of the programming language used for NVIDIA GPU's?

(5) (3) The roofline model creates a 2-dimensional graph. What are the two things that are being compared? (What are the values on the X and Y axis?)

(6) (3) How many entries are there in a (6,3) Gshare branch predictor? How many bits?

(7) (3) What is the primary difference between Tomasulo's algorithm and Scoreboarding?

(8) (3) The book lists several things that limit the amount of achievable ILP. List 3 of them.

Short Answers:

(9) (8) Understanding the hardware can influence how you write programs. Give at least 2 examples of how you might write software differently for a heavily pipelined machine verses a non-pipelined one, assuming performane (response time) matters. Be sure to explain why your modified software will be faster.

(10) (8) The book states that slow and wide architectures can be more power efficient than fast and narrow architectures. Explain why. Also, explain the underlying assumption that is being made, and why it is that we are still making narrow fast machines.

(11) (6) There are at least two types of control flow changes that standard dynamic branch predictors have trouble with. There is a technique that works well for one of these types ... name the two types of branches, and the technique used to successfully deal with one of them.

(12) (8) What is the definition of a basic block? Why is there a desire to create larger ones? Give one example of a way to create a bigger basic block.

(13) (8) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain what a precise interrupt is, why it is a challenge in OOO machines, and describe the main technique used today to provide precise interrupts.

(14) (6) The authors list 3 main hardware approaches to supporting multithreading. Briefly describe each one and how it works.

(15) (6) Register renaming is used to avoid name hazards. Is there a technique that can be used to avoid true (RAW) hazards? If so, what is it and how does it work? If not, why not?

(16) (8) Compare and contrast Superscalar and VLIW. Describe each, and list two advantages and one disadvantage of each approach.

(17) (6) Briefly outline how a Vector machine works, and what type of parallelism it is exploiting.

(18) (16) You are given the following code sequence:

```
ADDF    F2,F2,F4
SUBF    F2,F3,F2
MULTF   F3,F4,F2
DIVF    F2,F2,F3
```

Assume there are 8 logical and 16 physical registers. On the left below is the register mapping upon entering the code sequence. Your job is to fill in the mappings after the execution of the DIVF instruction, including what is on the free list. (Assume that during the execution of this code, no registers are released - in other words, the free list will be shorter at the end than at the beginning.)

BEFORE	
Logical	Physical
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	0

Free Pool: 1,3,5,7,9,11,13,15

AFTER	
Logical	Physical
0	
1	
2	
3	
4	
5	
6	
7	

Free Pool:

Now, rewrite the code sequence below using the actual physical register names instead of the logical ones.

```
ADDF   P__,P__,P__
SUBF   P__,P__,P__
MULTF  P__,P__,P__
DIVF   P__,P__,P__
```