1.)
a.)
Algorithm 1:
Init has 1 branch, 3 shifts/arithmetic
Loop has 1 load, 1 branch, and 3 shift/arithmetic, but is done 20 times.

CPU 1: Average is always **1 CPI**
CPU 2: Average is (2 + 3*4 + 20 * (5 + 2 + 3* 4)) / (4 + 20*5)
= (14 + 20*19) / (4 + 100)
= (14 + 380) / 104
= 394 / 104
= **3.788 CPI**

Algorithm 2:
Init has 2 arithmetic
Loop has 24 arithmetic/shifts and 1 branch
Branches based off of $7 != $9 $7 starts as `0x0a00 / 2560` `$9 as 0x5a00 / 23040`
$7 is incremented by `2560`
(23040 – 2560) / 2560 is 8 so loop done 8 times

CPU1: Average is always **1 CPI**
CPU 2: Average is (8 + 8*(24*4 + 2)) / (2 + 8*25)
= (8 + 8*(98)) / 202
= (8 + 784) / 202
= 792 / 202
= **3.921 CPI**

b.)
(Assuming CPUs are sequential)

I don't think I can adequately measure the preformance of the processors without knowing their clock rates as it effects the execution time of the programs. Say both had the same clock rate of 1 hz, then the first design would preform the algorithms above faster. The first does them in 104 and 202 seconds and the second in 394 and 792 second, nearly 4 times as fast. However, if the second CPU has a clock rate 4 times faster say 4 hz, then it will preform the programs in 98.5 and 198 seconds, a little faster than the first cpu with 1 hz.

For the team to determine which CPU has better preformance, they should measure how many times each program will be needed to execute and measure how long each CPU takes for each program. Then, they can weigh the times for each program relative to how many times they are needed to see which CPU gets done with a whole sequence of programs done faster instead of just the ones isolated.

For the CPUs to have the same preformance the second CPU needs to have a much higher clock rate. The second CPU needs to have a 1:3.788 clock rate difference compared to the first cpu for the first algorithm and a 1:3.921 for the second algorithm. This means when the first CPU has a clock rate of 1000 hz then the second needs to have 3788 hz and 3921 hz, per program. (Average is 3855 hz)

2.)
https://hackaday.com/2019/11/27/the-golden-age-of-ever-changing-computer-architecture/

The article talks about different chips being developed and says how new chips may be reaching the end of moores law. However, custom architechures for specific purposes may help improve the speed of these chips. They talk about google's TPU for AI and how it has specific instructions for matrix multiplication and specific deterministic behavior for AI. Next, they talk about a giant chip with 1.2 trillion transistors developed by cerebras with optimized operations for data processing.

This article relates to the ideas of instructions, execution time, and speculative execution that we have talked (or will talk) about in class. These processors are trying to minimize the execution time / preformance it takes for AI operations to occur. The way they are doing it is through optimized hardware solutions that a single specialize operation takes to preform complex matrix math. As well, these processors are attempting to predict what operations the AI will be doing next to speed up data processing the AI will undertake.