Question 1

1.1) $\frac{80}{100}(50) + \frac{10}{100}(50) + \frac{10}{100}(150)$

$= 8 + 5 + 15$

$= 28$ ns ✓ 2

1.2) $\frac{100}{100}(150)$

$= 150$ ns ✓ 1

1.3) Each processor can store and work on its own variables ~~instead of fetching it~~ and fetch them a lot quicker

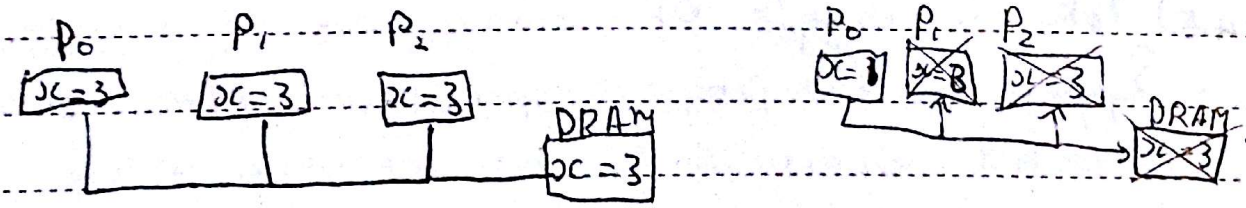
But if a processor edits a shared variable then it must update all the other processors for them to use it X 0

(some for distr. sys.)

Question 2

2.1) Each process has its own local copy of a shared variable. If one process modifies that variable it must go to each process and mark their copies as invalid and once they need to access their copy, the value must be updated. ✓ 2

2.2)



Now if P_0 modifies x to 1, P_1 and P_2 and DRAM have invalid copies of x. So P_0 goes to P_1 and P_2 and DRAM and marks their copies as invalid. ✓ (update?) 2

Question 4

4.1) The degree of concurrency is how many tasks can be executed simultaneously.
This will be limited to amount of processing elements available, as anything beyond that will be scheduled and will thus have an impact on the running time. ✓ 3

4.4) Take the example of the tile sliding puzzle.
Each move can be done in parallel as none of the moves depend on each other.
The performance can be drastically affected if the solution is very far into the ~~solution~~ state-space. Also the completely wrong solutions will be explored even when its obvious that that ~~the~~ particular branch will not give the right answer. ✓ (incomplete) 1

4.5) Take the example of finding the inverse of a matrix.
Suppose ~~an~~ the program speculates that it is a valid matrix and starts to compute the inverse but finds only near the end that the determinant is 0. All that computation will be wasted time. ✓ 2

Question 3.1

$$\int_5^{50} e^x \, dx = e^x \Big|_5^{50}$$

$$= e^{50} - e^5$$

$$= 5,18 \times 10^{21}$$

$$\frac{x}{5,18 \times 10^{21}} = 0,001\%.$$

$$x = 5,18 \times 10^{21} \times 0,001\%.$$

$$= 5,18 \times 10^{16}.$$

Question 4.2.

Using the Riemann Sum solution program, as the number of threads is increased to its max processing elements, the time to compleve the solution decreases. As it rises over the number of processing elements, the time to compleve the solution ~~is~~ increases.

Question 4.3

Take the Rieram Sum Solution.
The work could be distributed by 1 to $\frac{N}{2}$ and $\frac{N}{2}$ to N.
But the computation ~~get~~ takes longer as the number grows. So a more balanced work load would be all even number iterations and all odd number iterations. This way the work done between the 2 threads is distributed more fairly.

Question.....
Write on both sides of the paper

Do not write in this margin

1.4 (first) 2

3.1 0

3.2 0

4.2 0

4.3 0