15/16 HPC

1a. task parallelism is where a number of independent tasks are performed to achieve a common goal, data parallelism is where data is split into subsets and each subset is acted on to process the data faster than possible as a whole

b. Typed – each variable type must be declared, single datatypes are normally used but can handle mixed datatypes, fixed and variable length arrays

synchronous – when the sender or receiver reaches a point in the program where a message is sent they stop there and wait for the other to reach that point before continuing

ci. two typed channels are defined, chan 1 and chan 2, an int x is declared, 2 is sent from chan 1 and chan 2 receives a variable to be stored in x, then int s is declared and the next sequence 4 is sent on chan 2 and chan 1 receives a variable to store in s

ii. the sequences should run in parallel and chan 1 should output 2 and store 4 in s, and chan 2 should output 4 and then store 2 in x

d. In occam non-deterministic communication is introduced via the alt construct, an alt construct monitors several channels each of which is associated with a process, only the channel associated with the channel that first produces an input will be executed

e. software overhead - including index calculations necessitated by how the data were decomposed and assigned to processors

load imbalance – processors should be assigned the same amount of work or some will be idle while waiting for others

communication overhead – assuming communication and calculation cant overlap time spent communicating means less time available for processing

f. mutually exclusive access is where only one processor may access a variable at a time and no other process can access it until the first process has finished with it. This is achieved using locks, the first processor will lock the variable, perform its task and then release the lock so the variable can then be accessed by other processors.

2ai. This program is non deterministic as race conditions are created, there are 2 parallel sections that both try to access the same location in memory and seek to alter the data held there and there is no guaranteed order of execution.

ii. The program can be made deterministic by adding a lock to synchronise the sections, this way an order can be guaranteed and no error will be made by multiple processors reading or writing to the location at the same time.

iii.

bi.