

- d) What is the command to execute MPI program on the UNCC cluster `cci-gridgw.uncc.edu`? 2
- e) In the routine `MPI_Send(message,13,MPICHAR,x,10, MPI_COMM_WORLD)`, when can `x` be altered without affecting the message being transferred? 2
- f) A good implementation of the MPI routine `MPI_Barrier()` should be re-entrant. Why? 2
- g) Briefly what is the difference between a process and a thread? 2
- h) Suppose two concurrent threads execute the code sequence:
- ```
y = 1;
y ++;
x = y + 1;
```
- where `x` and `y` are shared variables. What are the possible values for `x` and `y` after both threads have completed. Clearly show how you got your answer. 2

i) What does the Paraguin compiler do? 2

j) What does the Paraguin directive `#pragma paraguin begin_parallel` do? 2

k) In OpenMP, several directives expect a structured block after the directive. Why is it not allowed to start that structured block with a `{` on the same line as the directive? 2

l) What does the following do?

```
forall (i = 1; i <= 10; i++) A[i]= B[i-1];
```

2

m) In Assignment 4 (Suzaku framework), what is the `suzaku.o` file? 2

n) Show how to do matrix-vector multiplication using a pipeline pattern.

2

o) Using the CUDA equations:

```
int x = threadIdx.x + blockIdx.x*blockDim.x;
int y = threadIdx.y + blockIdx.y*blockDim.y;
int index = x + y * N;
```

for flattening 2-dimensional addressing (x, y) into a 1 dimension index as a basis, develop an equation to flatten a 3-dimensional addressing (x, y, z) into a 1 dimension index. (There is more than solution.)

2

Qu. 2 Sequential code for odd-even transposition sort is given below:

```
int temp, i, j;
for (i = 0; i < N; i++){
 if (i & 1 == 1){
 for (j = 2; j < N; j+=2)
 if (a[j] < a[j-1]) {
 temp = a[j-1];
 a[j-1] = a[j];
 a[j] = temp;
 }
 } else {
 for (j = 1; j < N; j+=2)
 if (a[j] < a[j-1]) {
 temp = a[j-1];
 a[j-1] = a[j];
 a[j] = temp;
 }
 }
}
```

*(If there are any mistakes in this code, correct them.)*

(a) Show how the code sorts the numbers 4 2 7 8 5 1 3 6.

4

*In the following, provide very clear explanation of how your programs work. If I do not understand the code, I will assume it is incorrect. It is not necessary to give complete program, just that part that performs the sorting.*

(b) Write an OpenMP program to implement odd-even transposition sort using  $N/2$  threads ( $N$  even).

(c) Write an MPI program to implement odd-even transposition sort assuming one number is stored in each process ( $N$  processes). (It may be easier to have each pair of processes send their numbers and both do the comparison.)

8