* Only relevant answers are graded.

Q1

a) How input data decomposition benefits parallel and distributed computing? Give Example.

b) Name two examples that uses exploratory and speculative decomposition respectively. Now explain why the other type of decomposition is not applicable in each case.

c) Why we need a cluster distributed file system like HDFS? Explain all benefits.

d) Suppose a HDFS instance Namenode reboots. Explain what is lost and how the lost items will be recovered.

Q2

a) Explain the purpose of each line of the following MPI code.

```
int one, two, number;
MPI_Comm_rank(MPI_COMM_WORLD, &one);
MPI_Comm_size(MPI_COMM_WORLD, &two);
if (one == 0) {
    number = -1;
    MPI_Send(&number, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
} else if (one == 1) {
    MPI_Recv(&number, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    printf("Process 1 received number %d from process 0\n", number);
}
```

Write MPI C code snippet where each nodes determines if its rank is odd or even. Even nodes send their rank to odd nodes and vice versa. All nodes print theirs received numbers. Do not write template code. Hint: if (x % 2) { /* x is odd */ }

Explain what the following program is doing. Write small sentences as labels and draw data structure with values.

```
6     int rank, size;
7     const int array_size = 5;
8     float data[array_size], ps = 0.0; z = 0.0;
9     MPI_Init(&argc, &argv);
10    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
11    MPI_Comm_size(MPI_COMM_WORLD, &size);
12    if (rank == 0) for (int i = 0; i < array_size; i++) data[i] = i + 1.0;
13    MPI_Bcast(data, array_size, MPI_FLOAT, 0, MPI_COMM_WORLD);
14    for (int i = 0; i < array_size; i++) ps += data[i];
15    MPI_Reduce(&ps, &z, 1, MPI_FLOAT, MPI_SUM, 0, MPI_COMM_WORLD);
16    if (rank == 0) printf("%.2f\n", z);
17    MPI_Finalize();
```