* Only relevant answers are graded.

Q1

a) Consider the problem of multiplying two n x n matrices A and B to yield a matrix C. What are the benefits of decomposing the solution to this problem in 4, 8 & 12 tasks? Name all decompositions.

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) How a client application copy a file 512MB file into HDFS? Explain all steps including those taken internally by HDFS.

d) Suppose you attached a BackupNode with Namenode. Explain steps that will update BackupNode in-memory database same as Namenode.

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 sum theirs received numbers and print total. 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();
```