Question 1

1. Define the inputs and the outputs of the Map and Reduce functions.
   Map's inputs should be metadata of the file (including URL, size, date….), and outputs should be URL and its size. Reduce ought to calculate and add each size of URL and return total size.

2. Define the inputs and the outputs of the Map and Reduce functions.
   Map (key = filepath, value = "URL, size, data…")
   For each host URL
   Return (host, size)
   Reduce (key = host, value = size)
   Foreach size in sizes:
   Totalsize += size
   Return Totalsize

Question 2

1. Define the input and the output of the Map and Reduce functions.
   We will define two Maps, one of them take A as input and return rows in order, another one take B as input and return columns in order. Reduce functions are aimed to calculate elements of A row respectively multiply elements of B column, and its sums are elements of matrix C. For example, the consequence of i row of matrix A multiply j column of matrix B is [i,j] element of matrix C.

2. Write Map and Reduce functions of the matrix-matrix multiplication.
   Map1(key = A, values = [n * n]a)
   Int I = 1
   For(I; I <=n; I ++)
   Return a_{I1} ……a_{In}
   Map2(key = B, values = [n * n]c)
   Int I = 1
   For(I; I <=n; I ++)
   Return c_{I1} ……c_{In}
   Reduce1(key = a,c , values = na, nc)
   $C_{ij} = a_{i1}*b_{i1} + a_{i2}*b_{i2} + \cdots\cdots + a_{in}*b_{in}$
   Return C_{ij}
   Reduce2(key = C , values = C_{ij})
   Return [n * n]C

3. What if the matrices A and B do not fit in the mappers' memory? (How would you implement your Map-Reduce program?)
   If it is a stackoverflow problem, we can modify configuration to adjust mappers' memory, or we can modify parameters about map and yarn about memory allocation.
   We can implement Map-Reduce program via Hadoop platform, Client submits

MapReduce job; Jobtracker coordinates the operation of jobs. jobtracker is a Java application. Tasktracker runs tasks after job division. tasktracker is also a Java application. A distributed file system (usually HDFS) is used to share job files among other entities.

Question 3

1. Define the input and the output of the Map and Reduce functions of K-Mean algorithm

   Map should read the centroid information generated in the last iteration and decide which cluster each data point should belong to. Reduce will calculate all the data points and return new centroids as input in next iteration.

2. Write Map and Reduce functions of the K-Means algorithm

```
Map(key = centroid, values = centroid_id, K)
        Foreach object in objects:
                Distance = object.distance(centroid_id)
        If (Distance is least):
                Object belong to centroid
 Reduce(key = centroid, values = centroid_id, K)
        For I in range(k):
                For object in cluster:
                        Total +=object
                newCentroid = total/(n/k)
        return newCentroid
```