Assignment 1

We are interested in computing a math series sum of the type:
0.^t + 1.1^t + 2^t + ... + (n-1)^t
using client-server computing.

The value of n could be 100, 200, 300, 400, 500, but t value is a fixed non-negative integer.
The purpose is to distribute the task to 5 servers and compute the final result at the client. Client sends t (exponentiation) and p (partition number). Each server computes the result for its partition and sends the same to the client. Client adds the results returned by server 1, server 2, server 3, server 4 and server 5 and prints the final result.

We assume that n is evenly divisible by 5 (number of available servers).
Servers and gets the corresponding result

((n/5)p)^t + ((n/5)p + 1)^t + ... + ((n/5)(p+1)-1)^t

where p = 0, 1, 2, 3, 4

Layout for the request packet is:

1. Request/Reply
2. Function number
2. Partition number
3. range1 --> starting range of term
4. range2 --> ending range of terms
5. n --> total number of terms
6. t --> exponentiation

The layout of reply packet sent by server:

1. Request/reply
2. Function number
3. Partition number
4. Number of terms
5. n --> total number of terms
6. t --> exponentiation.
7. sum of partition.
8. I/X --> include/exclude terms
8. T_{(n/p)p}
9. T_{(n/p)p+1}
...
... T_{p[(n/5)+1]-1}

The server after receiving client request must decode it to find:
1. Partition number allocated to it.
2. Compute its partition sum
3. Encode the result
4. Include values of n, t and p.
5. Send the result to the client
The client after receiving a server's result stores it as the corresponding partition sum.
It waits until all servers have sent their corresponding results. Then adds the partition sum and prints the total sum, number of terms, exponentiation.

Use UDP socket to implement the problem.