PSEUDOCODE

1. Import necessary modules (mpi4py, math)

2. Define a function to check if a number is prime:
    1. If the number is less than or equal to 1, return False
    2. If the number is 2, return True
    3. If the number is divisible by 2, return False
    4. Iterate through odd numbers starting from 3 up to the square root of the number:
        1. If the number is divisible by any of these odd numbers, return False
    5. If none of the conditions above are met, return True

3. Define a function to find the maximum prime number within an interval:
    1. Initialize a variable 'mx' to 0
    2. Iterate through each number in the specified interval:
        1. If the number is prime and greater than 'mx', update 'mx' to be that number
    3. Return 'mx'

4. Define the main function to distribute workload, gather results, and print maximum prime numbers:
    1. Initialize MPI communicator, rank, and size
    2. Iterate through each value of N in the specified range:
        1. Calculate the interval size for the current processor
        2. Calculate the start and end points of the interval for the current processor
        3. Adjust the end point if it's the last processor and N is not evenly divisible by the number of processors
        4. Find the maximum prime number within the interval
        5. Gather all maximum prime numbers to the root processor
        6. If the current processor is the root processor, find the overall maximum prime number and print it along with the value of N

5. If the script is executed directly:
    1. Define start and end values for the range of N
    2. Iterate through each end value in the list:
        1. Print a header indicating the current range of N
        2. Call the main function with the start and end values

End of pseudo code.