CODE

DistributedSum.py

```
from mpi4py import MPI
import numpy as np

def distribute_elements(array, comm):
    n = len(array)
    local_n = n // comm.Get_size()  # Compute the local size of the array
    local_array = np.empty(local_n, dtype=int)  # Create an empty array to store the local elements
    comm.Scatter(array.tobytes(), local_array, root=0)  # Scatter the elements from the root process to all processes
    local_sum = np.sum(local_array)  # Compute the local sum of the elements
    return local_sum

if __name__ == '__main__':
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()  # Get the rank of the current process

    if rank == 0:
        n = 100  # specify the total number of elements
        array = np.arange(1, n + 1)  # Generate an array of numbers from 1 to n
    else:
        array = None

    array = comm.bcast(array, root=0)  # Broadcast the array from the root process to all processes

    local_sum = distribute_elements(array, comm)  # Compute the local sum of elements for each process

    all

    if r
        t
        ]
```

OUTPUT