

## Ex7p04

For this exercise I wrote the program p04.cpp. Below is the main logic of the program:

```
int sum = world_rank;
for (int stride = 1; stride < world_size; stride *= 2) {
    if ((world_rank / stride) % 2 == 0 && world_rank +
stride < world_size) {
        int receivedValue;
        MPI_Recv(&receivedValue, 1, MPI_INT, world_rank +
stride, 0,
                MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        sum += receivedValue;
    } else if ((world_rank / stride) % 2 == 1 && world_rank
- stride >= 0 ) {
        MPI_Send(&sum ,1 ,MPI_INT ,world_rank - stride
,0 ,MPI_COMM_WORLD);
        break;
    }
}
```

Each process checks every stride if it should send or receive data. If the process id divided by the stride is even and there is another process to its right, it receives a value and adds it to its sum. If it's uneven and there is a process to its left it instead sends its sum. The program works for all powers of two.

### Results:

*-np 4*

The total sum is: 6

*-np 8*

The total sum is: 28