

# REPORT - PW7

Nafila Amirli

---

## Exercise

Write an MPI program that allows a processor to communicate its rank around a ring. The sum of all ranks is then accumulated and printed out by each processor.

### Code

To get the neighbours(previous and next processes):

```
//identify ranks of neighbours
prev = rank - 1;
if (prev < 0)
{
    prev = prev + size;
}
next = (rank + 1) % size;
```

Receive buffer from previous process and send current rank to next, then sum all the numbers:

```
for (int i = 0; i < size; i++)
{
    //recieve from prev
    MPI_Irecv(&rBuf, 1, MPI_INT, prev, tag, MPI_COMM_WORLD, &requestR);
    //send to next
    MPI_Isend(&sBuf, 1, MPI_INT, next, tag, MPI_COMM_WORLD, &requestS);

    //wait receive to complete, then use rBuf to sum
    MPI_Wait(&requestR, status);
    sum = sum + rBuf;
    MPI_Wait(&requestS, status);
    sBuf = rBuf;
}
```

---

---

Terminal output:

```
nafile@nafile-Lenovo-V110-15ISK:~/paralel/PW7$ mpicc ex1.c
nafile@nafile-Lenovo-V110-15ISK:~/paralel/PW7$ mpiexec -n 5 ./a.out
proc 1 has sum = 10
proc 2 has sum = 10
proc 3 has sum = 10
proc 0 has sum = 10
proc 4 has sum = 10
MPI_Wtime measured : 0.090540
MPI_Wtime measured : 0.067271
MPI_Wtime measured : 0.070349
MPI_Wtime measured : 0.075155
```

With 4 processes:

```
nafile@nafile-Lenovo-V110-15ISK:~/paralel/PW7$ mpicc ex1.c
nafile@nafile-Lenovo-V110-15ISK:~/paralel/PW7$ mpiexec -n 4 ./a.out
proc 0 has sum = 6
proc 1 has sum = 6
proc 2 has sum = 6
proc 3 has sum = 6
MPI_Wtime measured : 0.000134
MPI_Wtime measured : 0.000039
MPI_Wtime measured : 0.000092
MPI_Wtime measured : 0.000118
```