**High Performance Computing Lab**

**Practical No. 6**

**PRN: 22510039**

**Batch: B2**

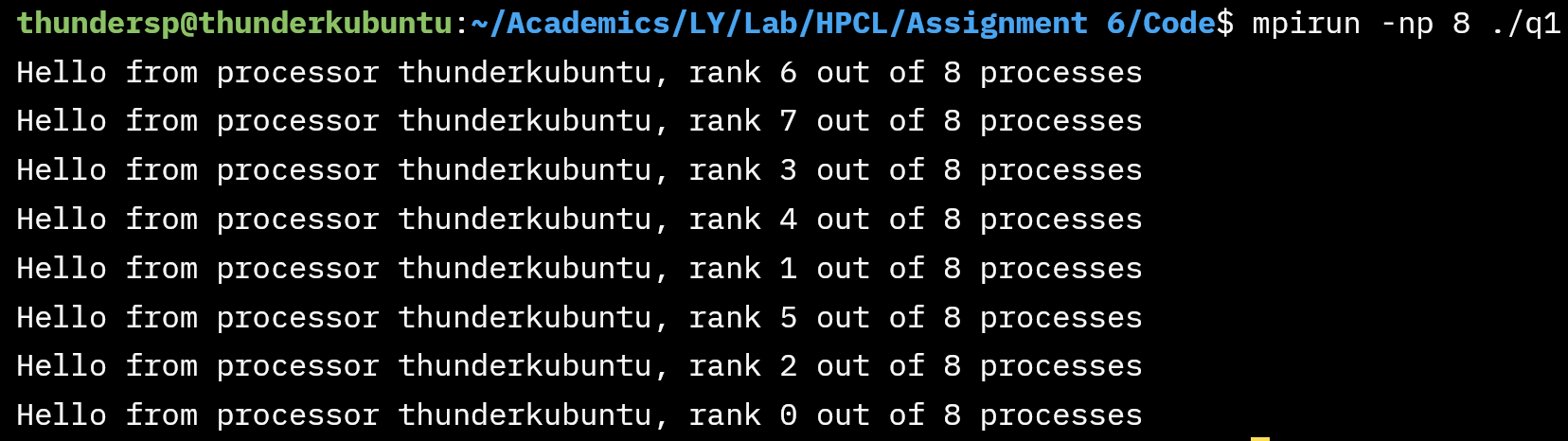
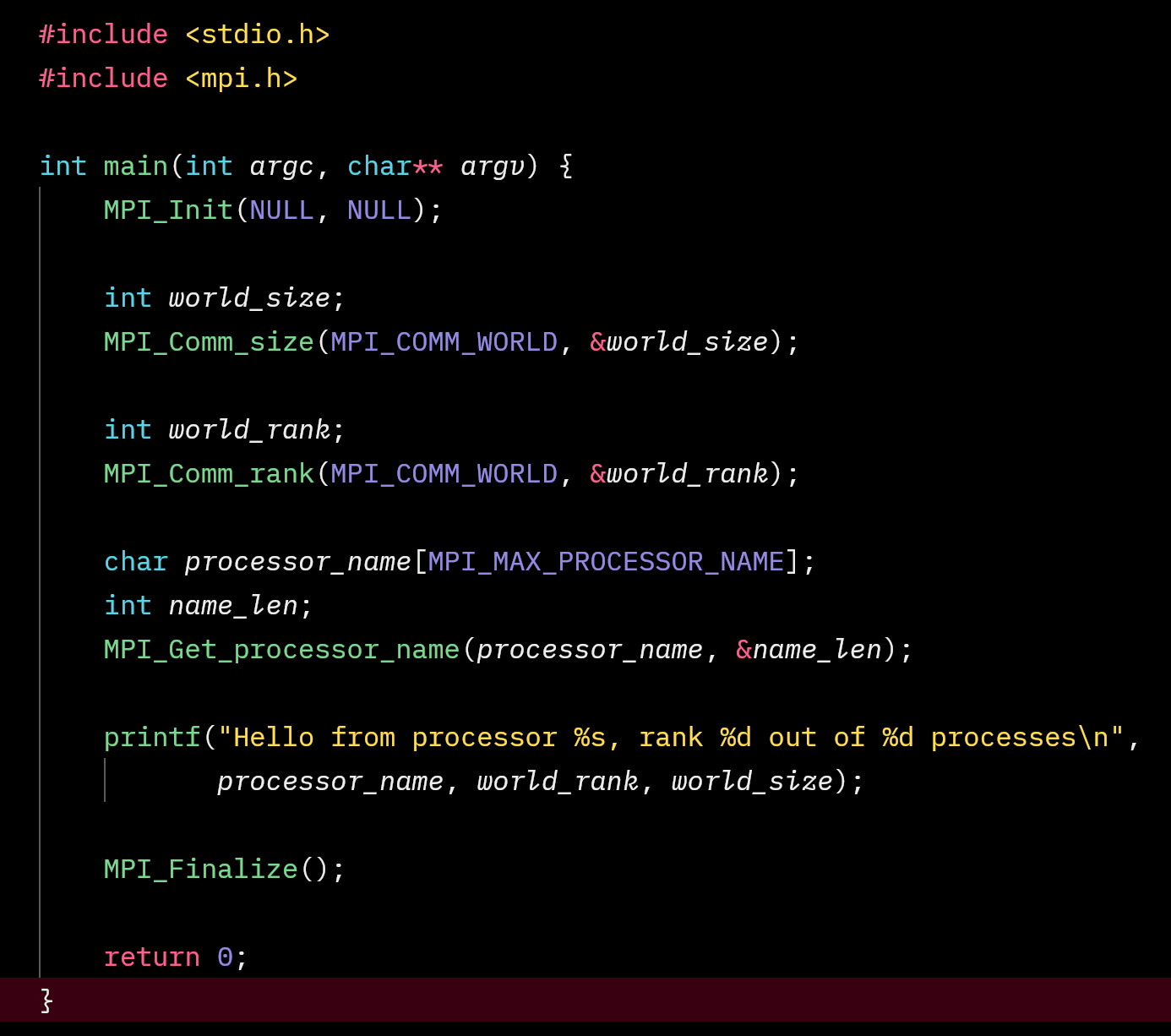
**Title of practical:**

Installation of MPI & Implementation of basic functions of MPI

**Problem Statement 1:**

Implement a simple hello world program by setting number of processes equal to 10

**Screenshots:**



**Information 1:**

To run the program successfully, we must specify the cores which are less than or equal to the max cores available in the system. If we specify more cores, it throws error unless we pass the parameter as oversubcribe.

Command to compile and run the code -

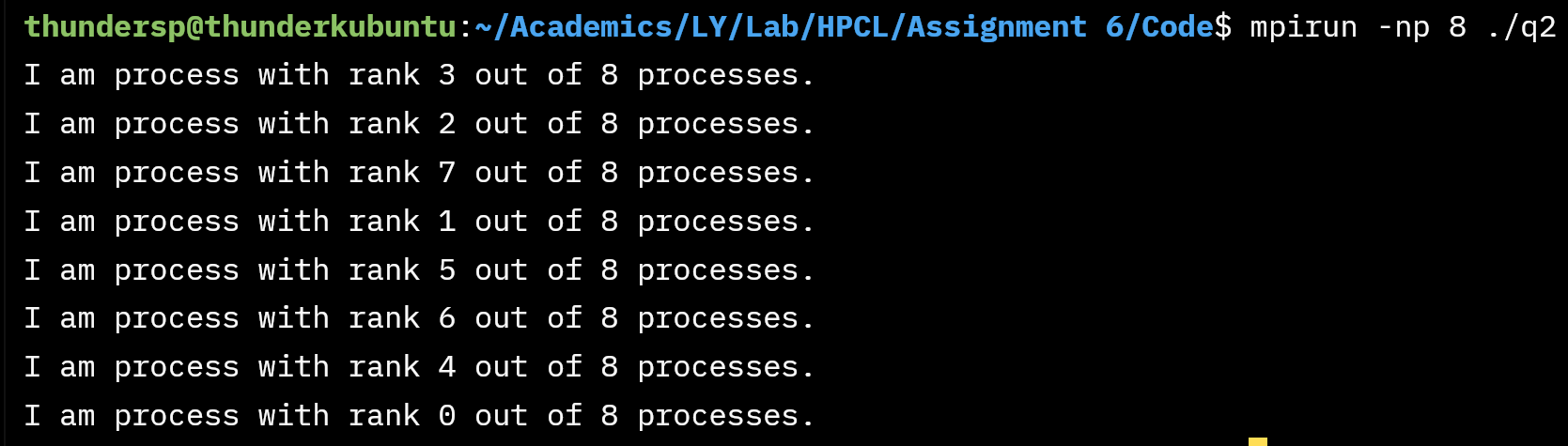
mpicc source –o output

mpirun –np 8 ./file

**Problem Statement 2:**

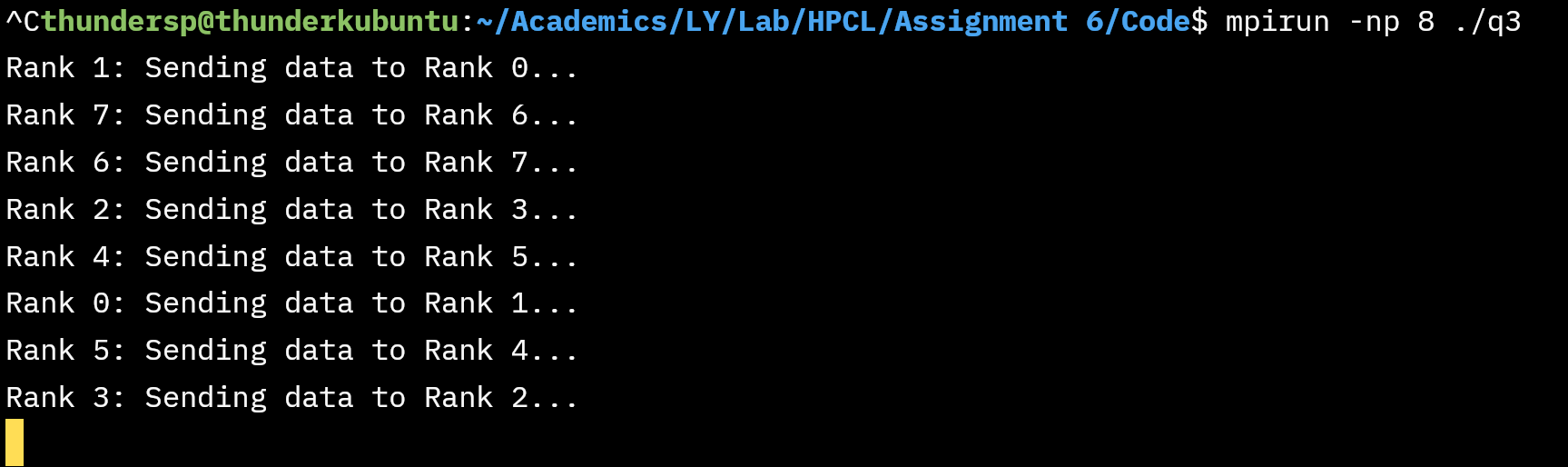
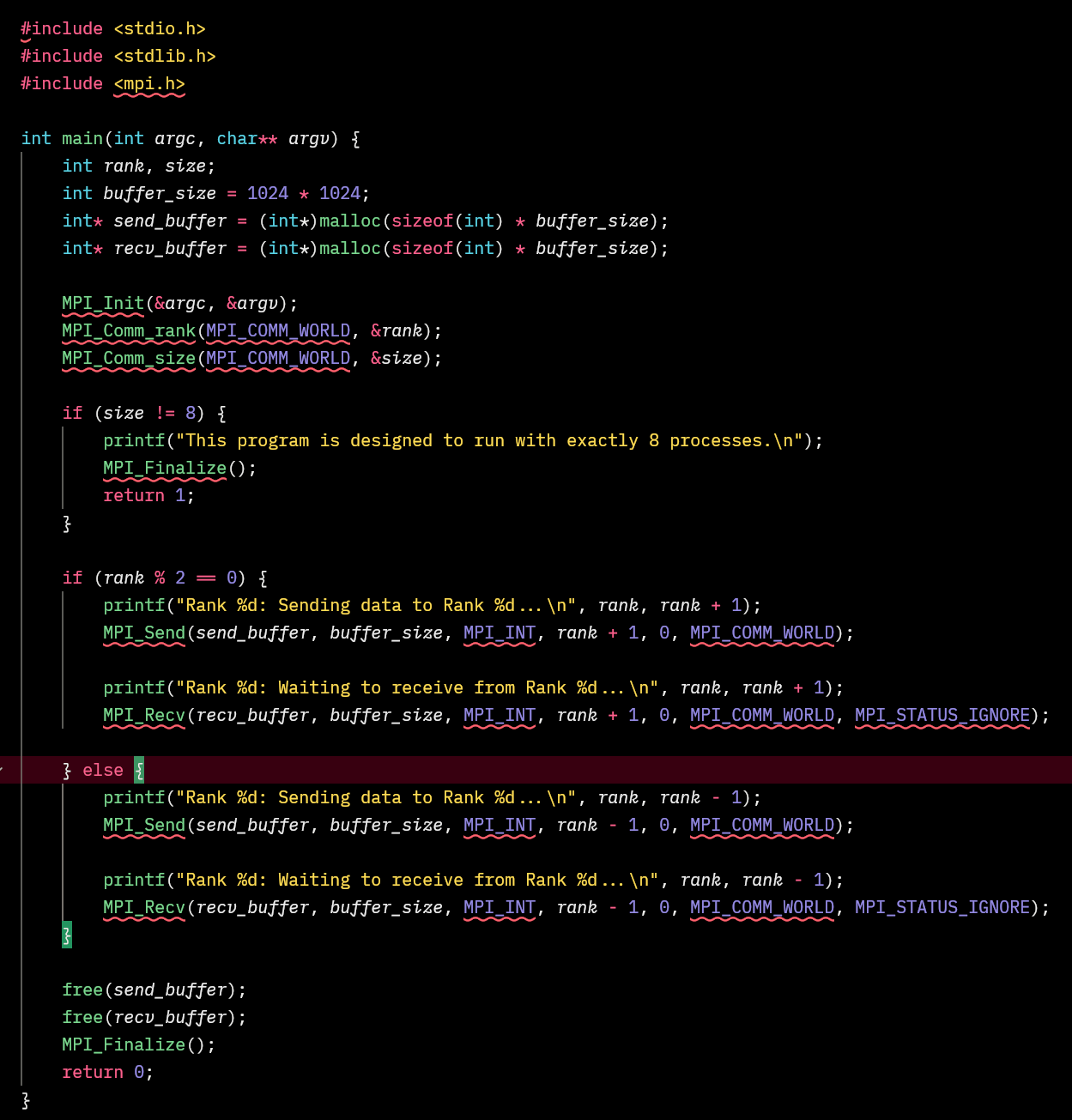
Implement a program to display rank and communicator group of five processes

**Screenshots:**



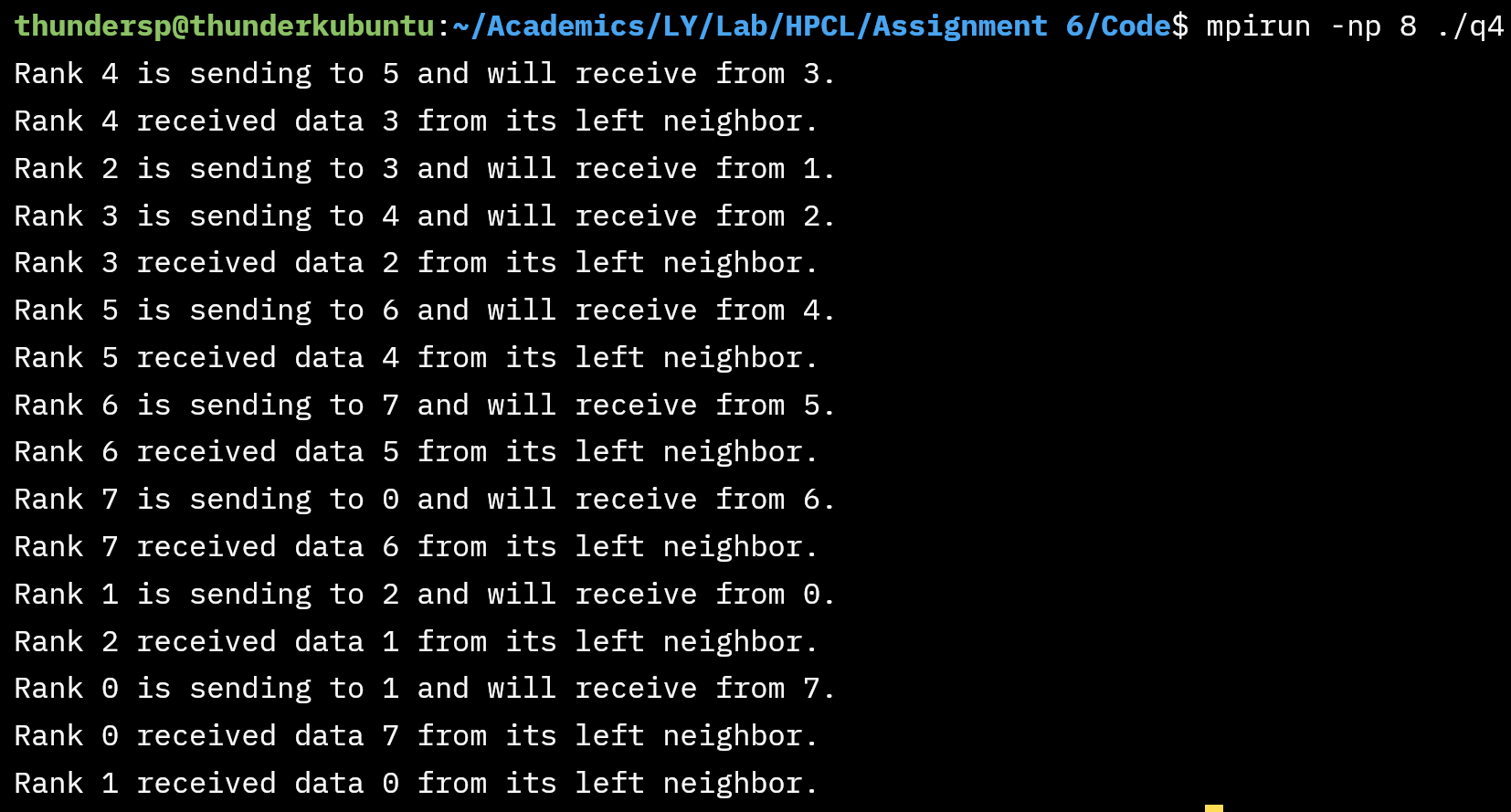
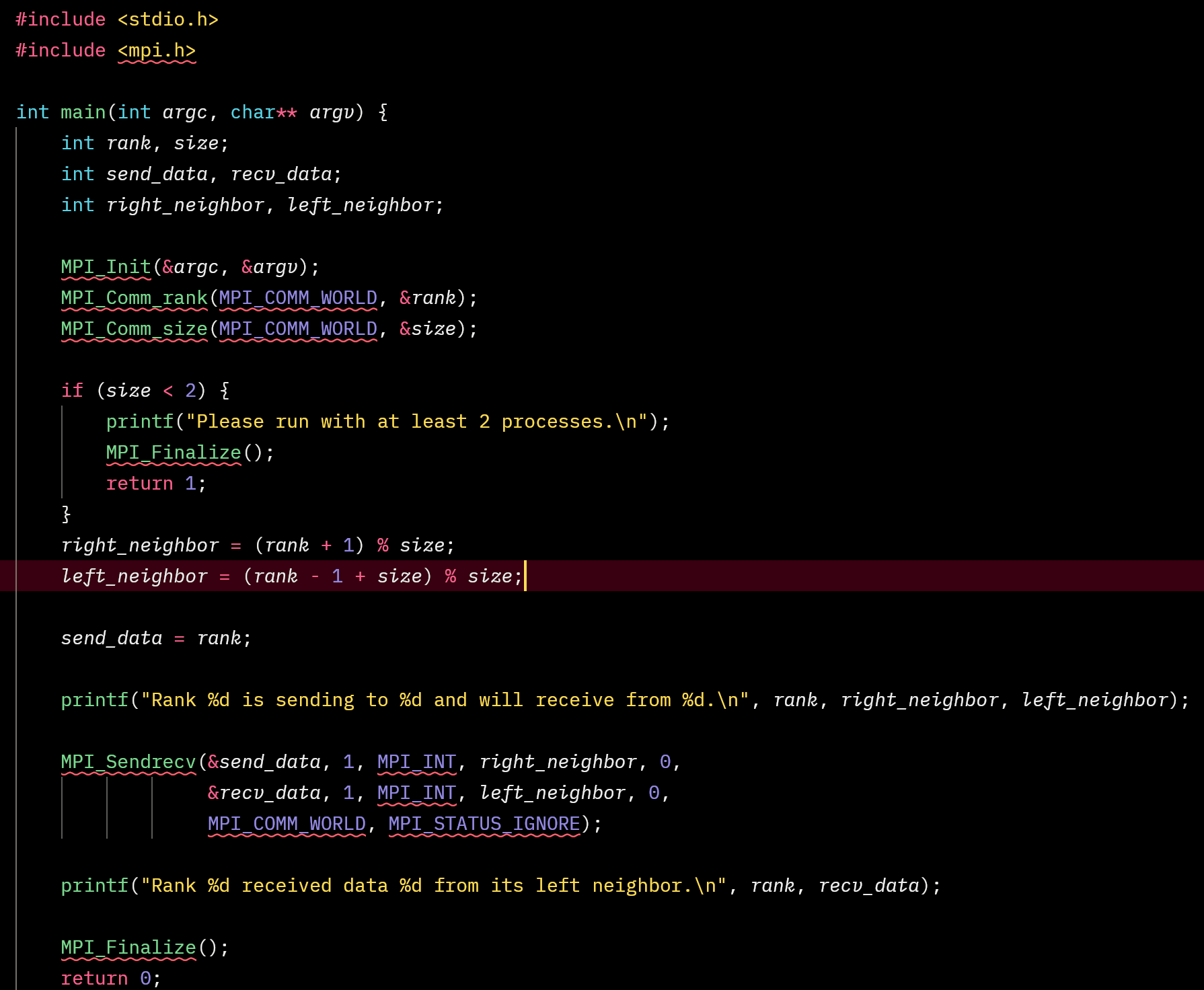
**Q3: Implement a MPI program to give an example of Deadlock.**

**Program and screenshots**



Program stops executing and infinitely stalls

**Q4. Implement blocking MPI send & receive to demonstrate Nearest neighbor exchange of data in a ring topology.**



**Q5. Write a MPI program to find the sum of all the elements of an array A of size**

**n. Elements of an array can be divided into two equals groups. The first [n/2]**

**elements are added by the first process, P0, and last [n/2] elements the by second process, P1. The two sums then are added to get the final result.**

