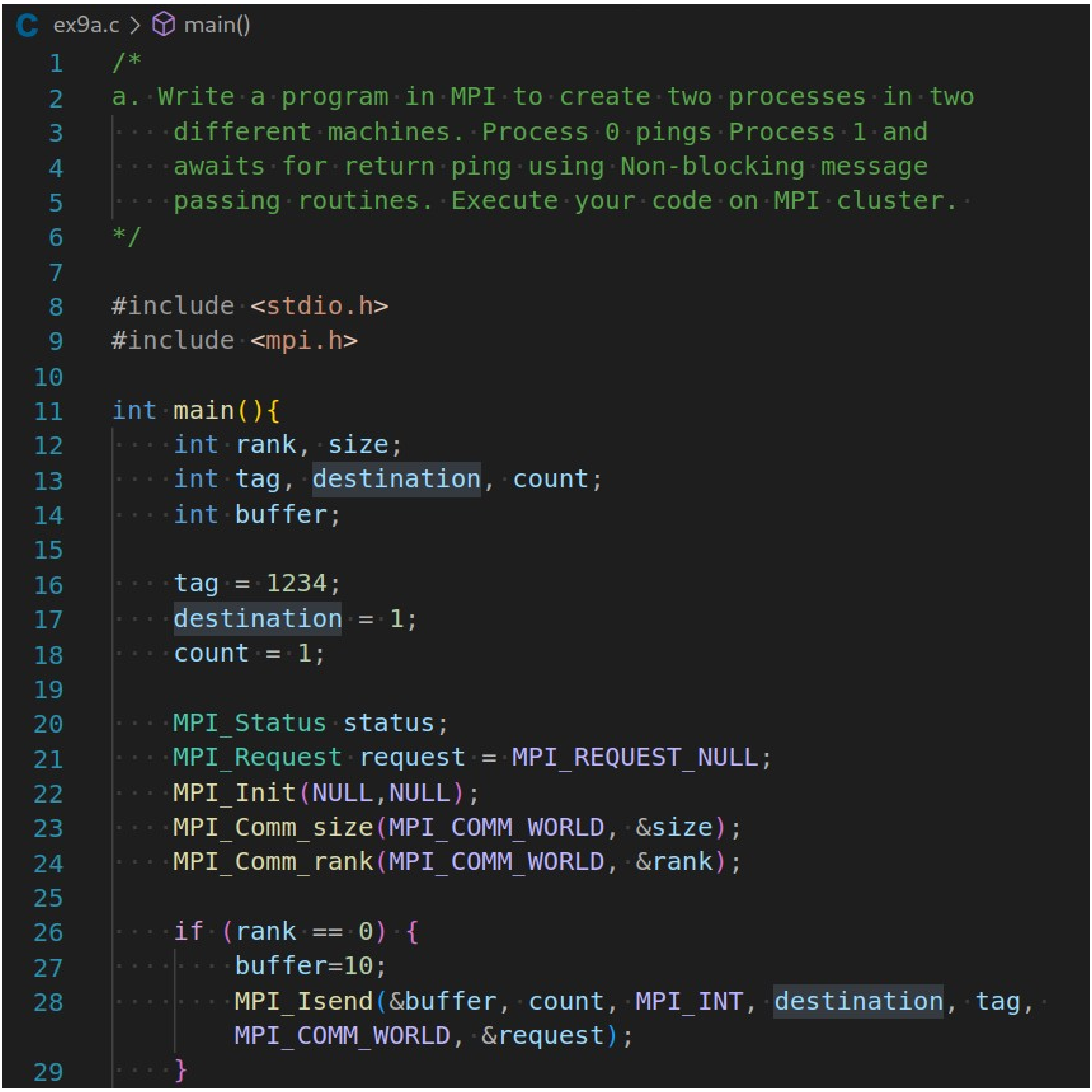
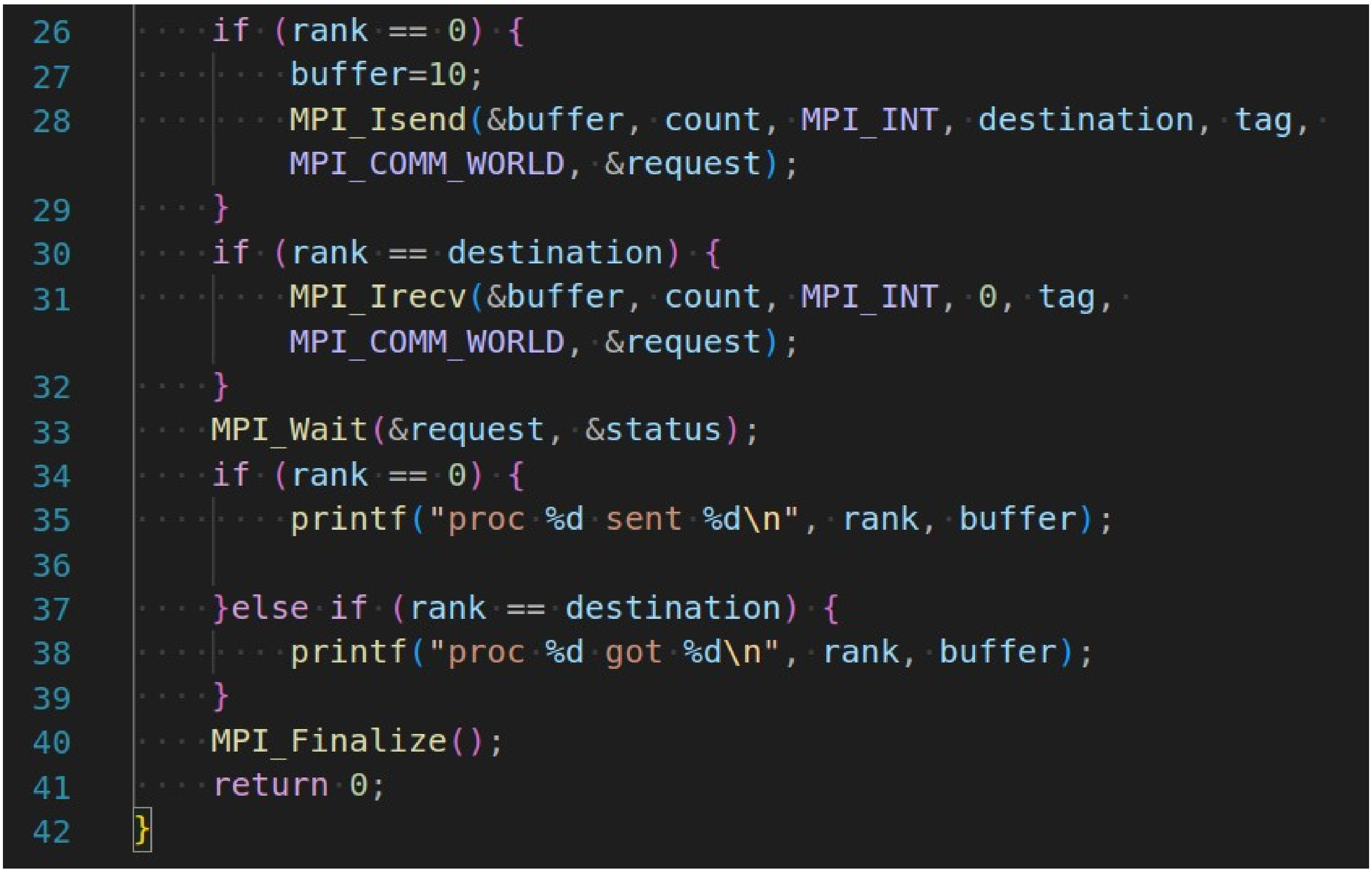
20BCE1550
Samridh Anand Paatni
CSE4001 Lab 9
MPI Point to Point Communication

a:

Code:

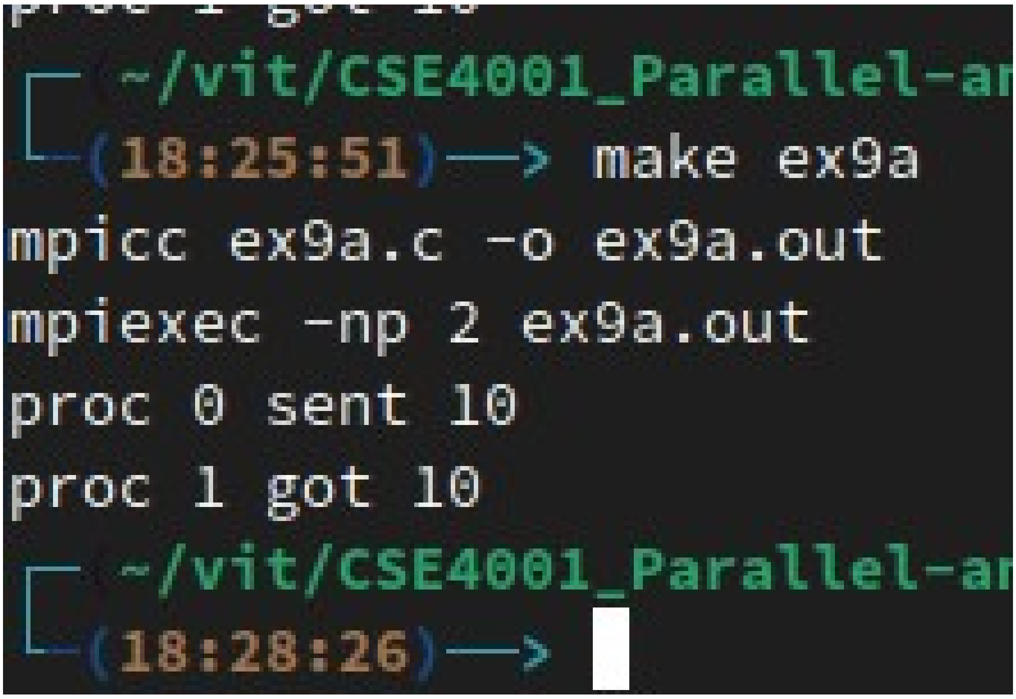


```c
/*
a. Write a program in MPI to create two processes in two
    different machines. Process 0 pings Process 1 and
    awaits for return ping using Non-blocking message
    passing routines. Execute your code on MPI cluster.
*/

#include <stdio.h>
#include <mpi.h>

int main(){
    int rank, size;
    int tag, destination, count;
    int buffer;

    tag = 1234;
    destination = 1;
    count = 1;

    MPI_Status status;
    MPI_Request request = MPI_REQUEST_NULL;
    MPI_Init(NULL,NULL);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    if (rank == 0) {
        buffer=10;
        MPI_Isend(&buffer, count, MPI_INT, destination, tag, MPI_COMM_WORLD, &request);
    }
```



```c
    if (rank == 0) {
        buffer=10;
        MPI_Isend(&buffer, count, MPI_INT, destination, tag, MPI_COMM_WORLD, &request);
    }
    if (rank == destination) {
        MPI_Irecv(&buffer, count, MPI_INT, 0, tag, MPI_COMM_WORLD, &request);
    }
    MPI_Wait(&request, &status);
    if (rank == 0) {
        printf("proc %d sent %d\n", rank, buffer);

    }else if (rank == destination) {
        printf("proc %d got %d\n", rank, buffer);
    }
    MPI_Finalize();
    return 0;
}
```

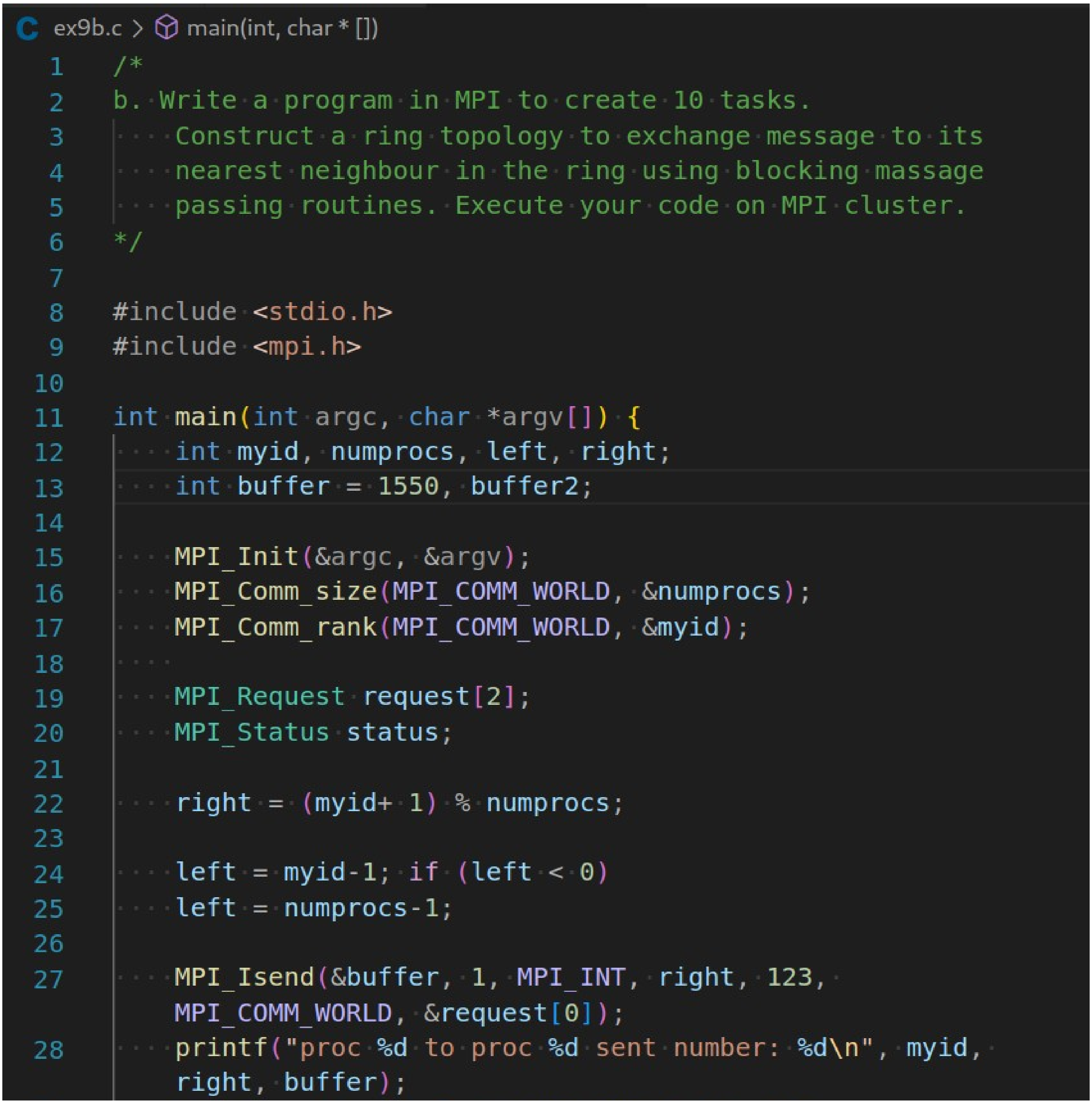
Output:



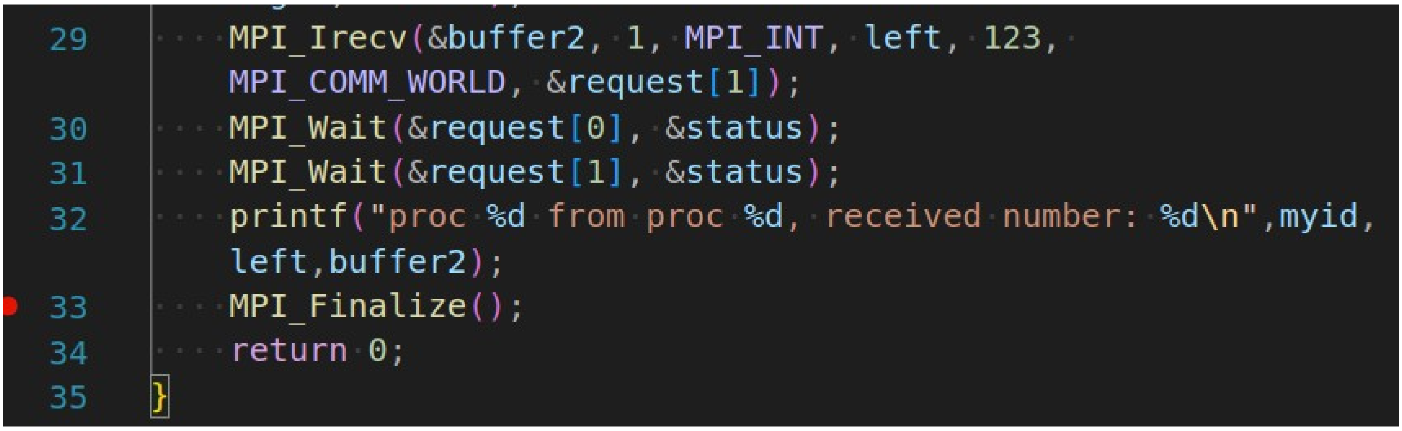
```
proc 1 got 10
┌─ ~/vit/CSE4001_Parallel-ar
└─(18:25:51)──> make ex9a
mpicc ex9a.c -o ex9a.out
mpiexec -np 2 ex9a.out
proc 0 sent 10
proc 1 got 10
┌─ ~/vit/CSE4001_Parallel-ar
└─(18:28:26)──>
```

b:

Code:

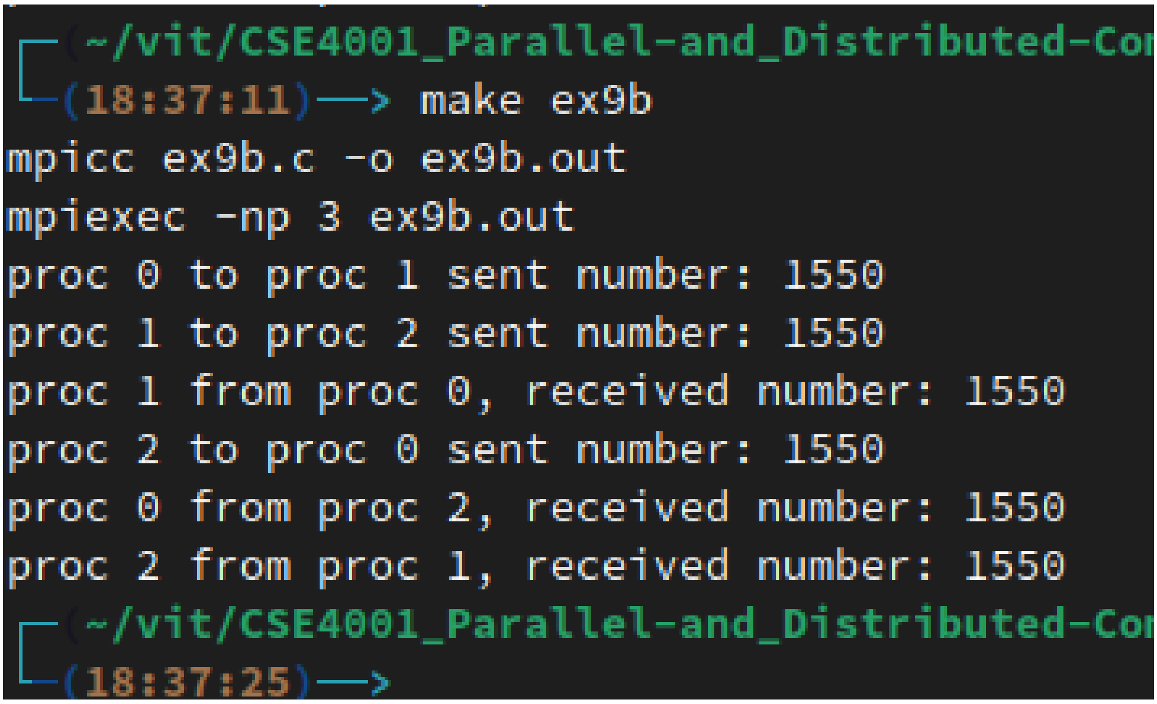


```c
/*
b. Write a program in MPI to create 10 tasks.
    Construct a ring topology to exchange message to its
    nearest neighbour in the ring using blocking massage
    passing routines. Execute your code on MPI cluster.
*/

#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[]) {
    int myid, numprocs, left, right;
    int buffer = 1550, buffer2;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    
    MPI_Request request[2];
    MPI_Status status;

    right = (myid+ 1) % numprocs;

    left = myid-1; if (left < 0)
    left = numprocs-1;

    MPI_Isend(&buffer, 1, MPI_INT, right, 123, MPI_COMM_WORLD, &request[0]);
    printf("proc %d to proc %d sent number: %d\n", myid, right, buffer);
```



```c
    MPI_Irecv(&buffer2, 1, MPI_INT, left, 123, MPI_COMM_WORLD, &request[1]);
    MPI_Wait(&request[0], &status);
    MPI_Wait(&request[1], &status);
    printf("proc %d from proc %d, received number: %d\n",myid, left,buffer2);
    MPI_Finalize();
    return 0;
}
```

Output:



```
┌─(~/vit/CSE4001_Parallel-and_Distributed-Co
└─(18:37:11)──> make ex9b
mpicc ex9b.c -o ex9b.out
mpiexec -np 3 ex9b.out
proc 0 to proc 1 sent number: 1550
proc 1 to proc 2 sent number: 1550
proc 1 from proc 0, received number: 1550
proc 2 to proc 0 sent number: 1550
proc 0 from proc 2, received number: 1550
proc 2 from proc 1, received number: 1550
┌─(~/vit/CSE4001_Parallel-and_Distributed-Co
└─(18:37:25)──>
```