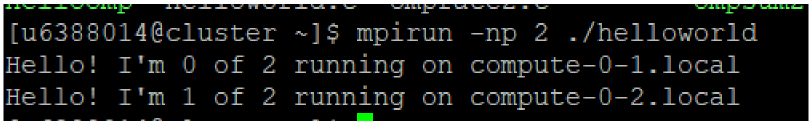
Lab 4: MPI Basics

Name: ID: Sec:

Save your file to “lab4_63xxxxxx.pdf” and upload it to MyCourses website.

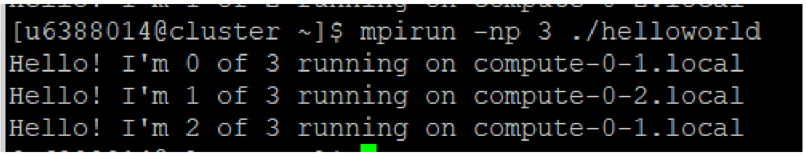
Q1. helloworld.c

Output from running with 2 processes:



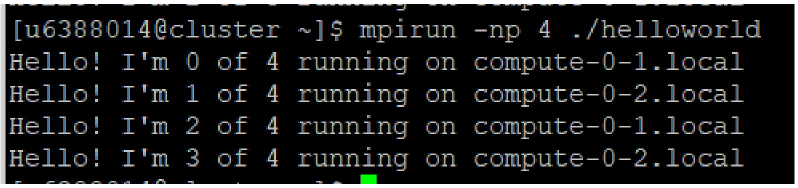
```
[u6388014@cluster ~]$ mpirun -np 2 ./helloworld
Hello! I'm 0 of 2 running on compute-0-1.local
Hello! I'm 1 of 2 running on compute-0-2.local
```

Output from running with 3 processes



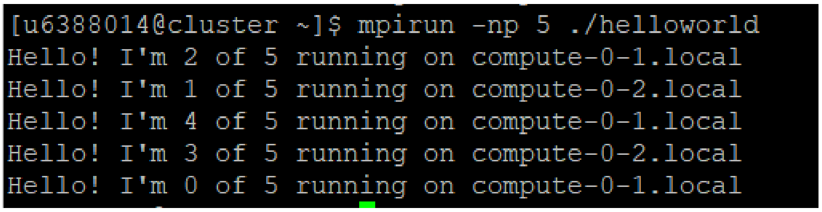
```
[u6388014@cluster ~]$ mpirun -np 3 ./helloworld
Hello! I'm 0 of 3 running on compute-0-1.local
Hello! I'm 1 of 3 running on compute-0-2.local
Hello! I'm 2 of 3 running on compute-0-1.local
```

Output from running with 4 processes:



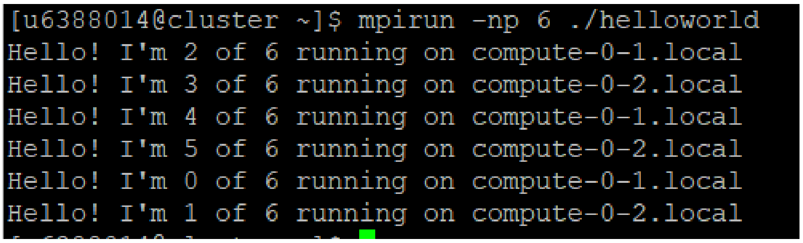
```
[u6388014@cluster ~]$ mpirun -np 4 ./helloworld
Hello! I'm 0 of 4 running on compute-0-1.local
Hello! I'm 1 of 4 running on compute-0-2.local
Hello! I'm 2 of 4 running on compute-0-1.local
Hello! I'm 3 of 4 running on compute-0-2.local
```

Output from running with 5 processes:



```
[u6388014@cluster ~]$ mpirun -np 5 ./helloworld
Hello! I'm 2 of 5 running on compute-0-1.local
Hello! I'm 1 of 5 running on compute-0-2.local
Hello! I'm 4 of 5 running on compute-0-1.local
Hello! I'm 3 of 5 running on compute-0-2.local
Hello! I'm 0 of 5 running on compute-0-1.local
```

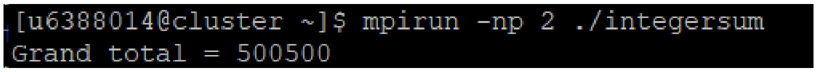
Output from running with 6 processes:



```
[u6388014@cluster ~]$ mpirun -np 6 ./helloworld
Hello! I'm 2 of 6 running on compute-0-1.local
Hello! I'm 3 of 6 running on compute-0-2.local
Hello! I'm 4 of 6 running on compute-0-1.local
Hello! I'm 5 of 6 running on compute-0-2.local
Hello! I'm 0 of 6 running on compute-0-1.local
Hello! I'm 1 of 6 running on compute-0-2.local
```

Q2. integersum.c

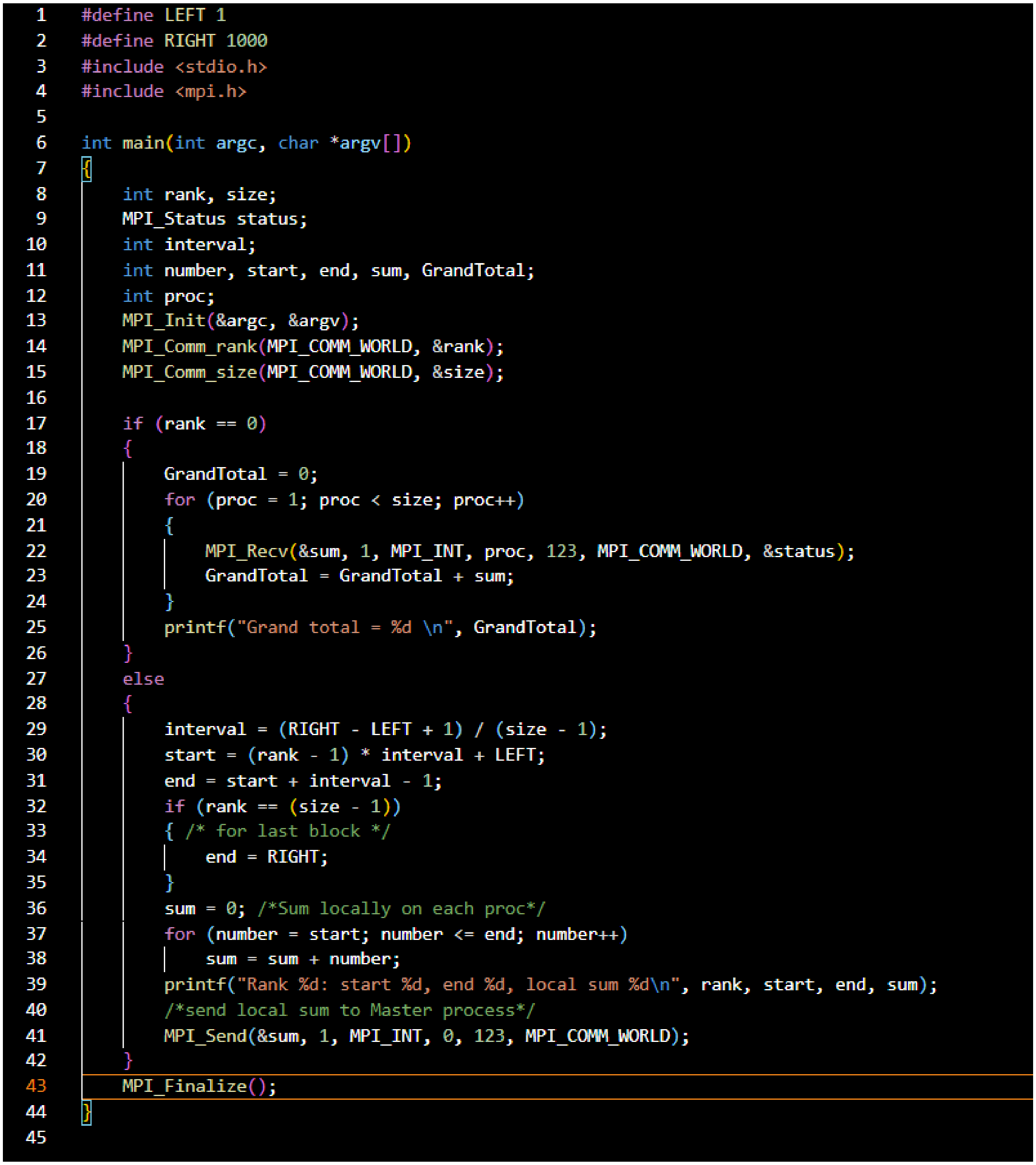
Output of integersum



```
[u63880l4@cluster ~]$ mpirun -np 2 ./integersum
Grand total = 500500
```

Q3. Modified integersum.c

Source Code:



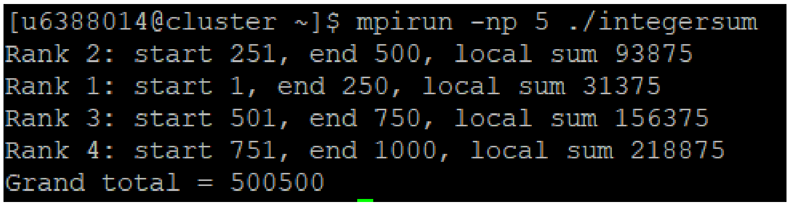
```c
#define LEFT 1
#define RIGHT 1000
#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[])
{
    int rank, size;
    MPI_Status status;
    int interval;
    int number, start, end, sum, GrandTotal;
    int proc;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank == 0)
    {
        GrandTotal = 0;
        for (proc = 1; proc < size; proc++)
        {
            MPI_Recv(&sum, 1, MPI_INT, proc, 123, MPI_COMM_WORLD, &status);
            GrandTotal = GrandTotal + sum;
        }
        printf("Grand total = %d \n", GrandTotal);
    }
    else
    {
        interval = (RIGHT - LEFT + 1) / (size - 1);
        start = (rank - 1) * interval + LEFT;
        end = start + interval - 1;
        if (rank == (size - 1))
        { /* for last block */
            end = RIGHT;
        }
        sum = 0; /*Sum locally on each proc*/
        for (number = start; number <= end; number++)
            sum = sum + number;
        printf("Rank %d: start %d, end %d, local sum %d\n", rank, start, end, sum);
        /*send local sum to Master process*/
        MPI_Send(&sum, 1, MPI_INT, 0, 123, MPI_COMM_WORLD);
    }
    MPI_Finalize();
}

```

Output



```
[u6388014@cluster ~]$ mpirun -np 5 ./integersum
Rank 2: start 251, end 500, local sum 93875
Rank 1: start 1, end 250, local sum 31375
Rank 3: start 501, end 750, local sum 156375
Rank 4: start 751, end 1000, local sum 218875
Grand total = 500500
```