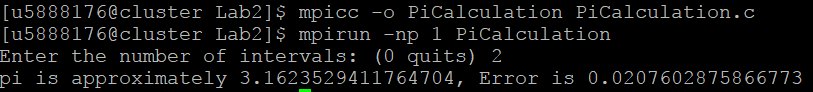
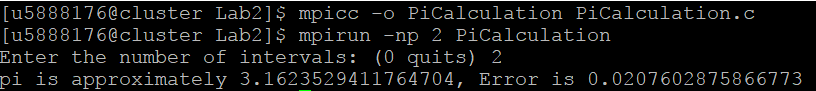
**Question1:** Capture the screen output from “Pi Calculation” Program with different number of intervals and number of processes.

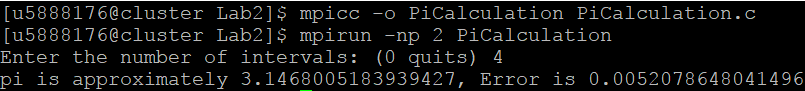
Output 1: 1 processes 2 intervals



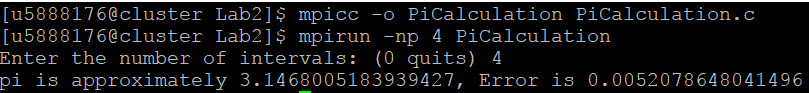
Output 2: 2 processes 2 intervals



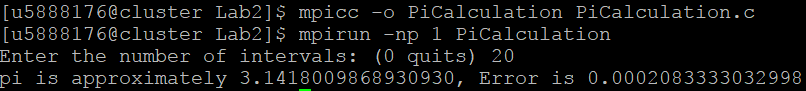
Output 3: 2 processes 4 intervals



Output 4: 4 processes 4 intervals



Output 5: 1 processes 20 intervals



**Question2:** Explain briefly, how the Pi value is computed by the program?

**Answer**: Firstly, it starts to broadcast to all processes. Then, it starts to divide the whole interval and sends each divided one to calculate in each process (using integral of 4/(1+x^2) from range 0 to 1). Lastly, I use MPI\_Reduce to collect all results in every process into one value to print the pi value out.

**Question3:** Place your “modified intergersum.c” program in the space below.

---- Place your code here ----

#include <stdio.h>

#include <mpi.h>

int main(int argc, char \*argv[])

{

int LEFT, RIGHT;

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) {

printf("Enter the number of LEFT and RIGHT: ");

fflush(stdout);

scanf("%d %d", &LEFT, &RIGHT);

GrandTotal = 0;

for (proc=1; proc<size; proc++) {

GrandTotal = GrandTotal+sum;

}

printf("Grand total = %d \n", GrandTotal);

}

MPI\_Bcast(&LEFT, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

MPI\_Bcast(&RIGHT, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

if(rank != 0){

interval = (RIGHT - LEFT + 1)/(size - 1);

start = (rank - 1)\*interval + LEFT;

end = start + interval-1;

if (rank == (size-1)) {

end = RIGHT;

}

sum=0;

for (number=start; number<=end; number++)

sum = sum+number;

printf("Rank: %d start %d, end %d, local sum %d\n", rank, start, end, sum);

MPI\_Reduce(&sum, &GrandTotal, 1, MPI\_INT, MPI\_SUM, 0, MPI\_COMM\_WORLD);

}

MPI\_Finalize();

}

---- Place your code here ----

**Question4:** What is the function you use for broadcasting “left” and “right”?

**Answer**: MPI\_Bcast(void \*buffer, int count, MPI\_Datatype datatype, int root, MPI\_Com\_World)

**Question5:** Which MPI operator is used in your “integersum.c”? What is the meaning of it? Hint: MPI\_Reduce function

**Answer**: I use MPI\_Reduce function as a MPI operator. MPI\_Reduce uses for reducing values on all processes into one value.