Slip 30

q.1

#include <stdio.h>

#include <stdlib.h>

#include <mpi.h>

#define ARRAY\_SIZE 1000

int main(int argc, char \*\*argv) {

int rank, size;

int array[ARRAY\_SIZE];

int local\_min,global\_min;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

srand(rank);

for (int i = 0; i < ARRAY\_SIZE / size; i++) {

array[i] = rand();

}

local\_min = array[0];

for (int i = 1; i < ARRAY\_SIZE / size; i++) {

if (array[i] < local\_min) {

local\_min = array[i];

}

}

MPI\_Reduce(&local\_min, &global\_min, 1, MPI\_INT, MPI\_MIN, 0, MPI\_COMM\_WORLD);

if (rank == 0) {

printf("Minimum number: %d\n", global\_min);

}

MPI\_Finalize();

return 0;

}

q.2

#include<stdio.h>

#include<stdlib.h>

int main()

{

int RQ[100],i,n,TotalHeadMoment=0,initial;

printf("Enter the number of Requests\n");

scanf("%d",&n);

printf("Enter the Requests sequence\n");

for(i=0;i<n;i++)

scanf("%d",&RQ[i]);

printf("Enter initial head position\n");

scanf("%d",&initial);

for(i=0;i<n;i++)

{

TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);

initial=RQ[i];

}

printf("Total head moment is %d",TotalHeadMoment);

return 0;

}