Q1

#include <stdio.h>

#include <string.h>

#include <mpi.h>

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

{

int rank, size, n;

MPI\_Status status;

MPI\_Init(&argc, &argv);

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

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

if (rank == 0)

{

char a[25];

fprintf(stdout, "Enter the Word : \n");

fflush(stdout);

scanf("%s", a);

n = strlen(a);

MPI\_Ssend(&n, 1, MPI\_INT, 1, 0, MPI\_COMM\_WORLD);

printf("Sent\n");

MPI\_Ssend(a, n + 1, MPI\_CHAR, 1, 1, MPI\_COMM\_WORLD);

fprintf(stdout, "Process %d sent: %s\n", rank, a);

fflush(stdout);

MPI\_Recv(a, n + 1, MPI\_CHAR, 1, 2, MPI\_COMM\_WORLD, &status);

fprintf(stdout, "Process %d recieved: %s\n", rank, a);

fflush(stdout);

}

else if (rank == 1)

{

char b[25];

MPI\_Recv(&n, 1, MPI\_INT, 0, 0, MPI\_COMM\_WORLD, &status);

printf("Got\n");

MPI\_Recv(b, n + 1, MPI\_CHAR, 0, 1, MPI\_COMM\_WORLD, &status);

fprintf(stdout, "Process %d recieved: %s\n", rank, b);

fflush(stdout);

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

{

if (b[i] >= 'a' && b[i] <= 'z')

{

b[i] = b[i] - 32;

}

else

{

b[i] = b[i] + 32;

}

}

MPI\_Ssend(b, n + 1, MPI\_CHAR, 0, 2, MPI\_COMM\_WORLD);

fprintf(stdout, "Process %d sent: %s\n", rank, b);

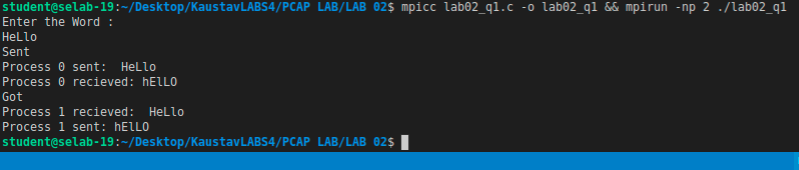
fflush(stdout);

}

MPI\_Finalize();

return 0;

}



Q2

#include "mpi.h"

#include <stdio.h>

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

{

int a, b, c;

int rank, size;

MPI\_Status status;

MPI\_Init(&argc, &argv);

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

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

if (rank == 0)

{

printf("Enter a value in the master process (rank %d) ", rank);

scanf(" %d", &a);

for (int i = 1; i < size; i++)

MPI\_Send(&a, 1, MPI\_INT, i, 1, MPI\_COMM\_WORLD);

}

else

{

MPI\_Recv(&b, 1, MPI\_INT, 0, 1, MPI\_COMM\_WORLD, &status);

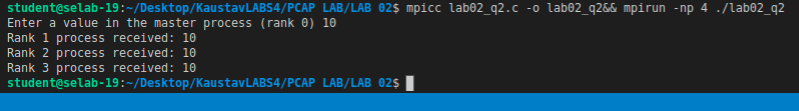
fprintf(stdout, "Rank %d process received: %d \n", rank, b);

}

MPI\_Finalize();

return 0;

}



Q3

#include "mpi.h"

#include <stdio.h>

#include <stdlib.h>

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

{

int x;

int rank, size;

MPI\_Status status;

MPI\_Init(&argc, &argv);

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

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

if (rank == 0)

{

int \*arr = (int \*)calloc(size, sizeof(int));

printf(" Enter %d number of values in the aray ", size - 1);

for (int i = 1; i < size; i++)

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

for (int i = 1; i < size; i++)

MPI\_Send(&arr[i], 1, MPI\_INT, i, i, MPI\_COMM\_WORLD);

}

else

{

MPI\_Recv(&x, 1, MPI\_INT, 0, rank, MPI\_COMM\_WORLD, &status);

if (rank % 2 == 0)

fprintf(stdout, "Even Rank %d process received: %d. After Squaring even ranked process : %d \n", rank, x, x \* x);

else

fprintf(stdout, "Odd Rank %d process received: %d. After Cubing odd ranked process : %d \n", rank, x, x \* x \* x);

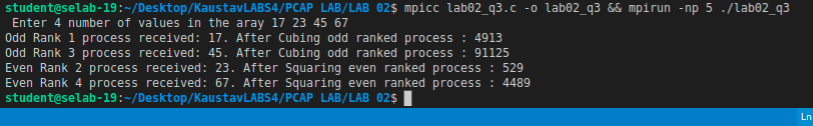
fflush(stdout);

}

MPI\_Finalize();

return 0;

}



Q4

#include "mpi.h"

#include <stdio.h>

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

{

int a, b, c;

int rank, size;

MPI\_Status status;

MPI\_Init(&argc, &argv);

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

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

if (rank == 0)

{

printf("Enter a value in the master process (rank %d) ", rank);

scanf(" %d", &a);

MPI\_Send(&a, 1, MPI\_INT, 1, size, MPI\_COMM\_WORLD);

MPI\_Recv(&a, 1, MPI\_INT, size - 1, size, MPI\_COMM\_WORLD, &status);

a = a + 1;

fprintf(stdout, "Rank %d process received: %d from process %d \n", rank, a, size - 1);

fflush(stdout);

}

else if (rank != size - 1)

{

MPI\_Recv(&b, 1, MPI\_INT, rank - 1, size, MPI\_COMM\_WORLD, &status);

fprintf(stdout, "Rank %d process received : %d from process %d \n", rank, b, rank - 1);

fflush(stdout);

b = b + 1;

MPI\_Send(&b, 1, MPI\_INT, rank + 1, size, MPI\_COMM\_WORLD);

}

else

{

MPI\_Recv(&c, 1, MPI\_INT, size - 2, size, MPI\_COMM\_WORLD, &status);

fprintf(stdout, "Rank %d process received: %d from process %d \n", rank, c, rank - 1);

MPI\_Send(&c, 1, MPI\_INT, 0, size, MPI\_COMM\_WORLD);

fflush(stdout);

}

MPI\_Finalize();

return 0;

}

