

**Walchand College of Engineering, Sangli**  
**Department of Computer Science and Engineering**

**Class:** Final Year (Computer Science and Engineering)

**Year:** 2022-23

**Semester:** 1

**Course:** High Performance Computing Lab

**Practical No. 5**

**Exam Seat No:** 2019BTECS00064

**Name –** Kunal Santosh Kadam

**Title of practical:** Installation of MPI and implementation of basic  
functions of MPI

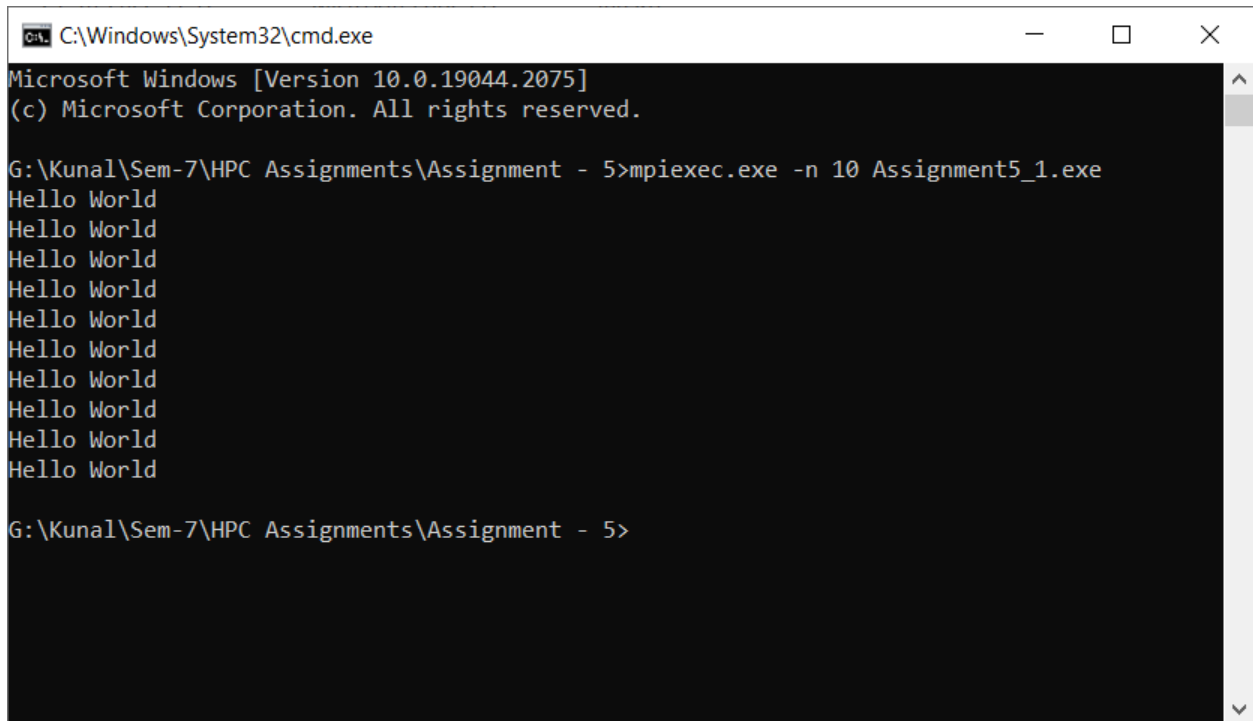
Complete the installation of MPI on the platform chosen by you

**Walchand College of Engineering, Sangli**  
**Department of Computer Science and Engineering**

**Problem Statement 1:**

Implement a simple hello world program by setting number of processes equal to 10

**Screenshot #:**

A screenshot of a Windows Command Prompt window. The title bar shows 'C:\Windows\System32\cmd.exe'. The window content shows the Microsoft Windows version (10.0.19044.2075) and copyright information. The user is at the prompt 'G:\Kunal\Sem-7\HPC Assignments\Assignment - 5>' and has entered the command 'mpiexec.exe -n 10 Assignment5\_1.exe'. The output shows ten lines of 'Hello World' printed in a light blue color. The prompt is now 'G:\Kunal\Sem-7\HPC Assignments\Assignment - 5>' again.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2075]
(c) Microsoft Corporation. All rights reserved.

G:\Kunal\Sem-7\HPC Assignments\Assignment - 5>mpiexec.exe -n 10 Assignment5_1.exe
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World

G:\Kunal\Sem-7\HPC Assignments\Assignment - 5>
```

**Information #:**

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

/* run this program using the console pauser or add your own getch,
system("pause") or input loop */

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

**Walchand College of Engineering, Sangli**  
**Department of Computer Science and Engineering**

```
//Initialize the MPI environment
MPI_Init(NULL,NULL);

printf("Hello World\n");

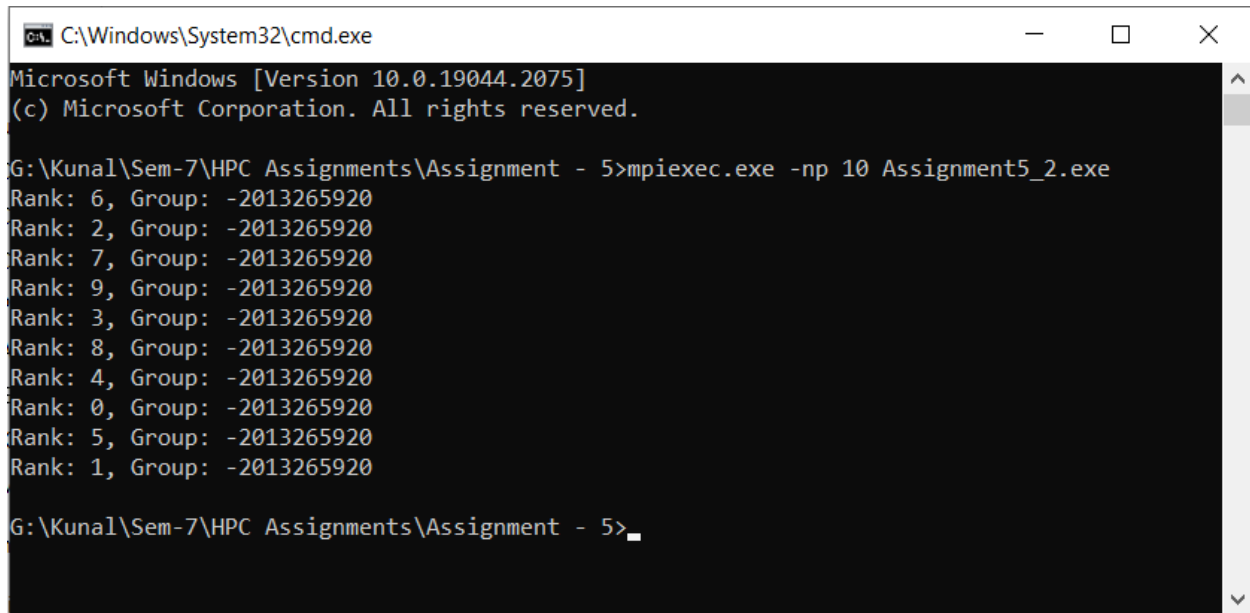
//Finalize the MPI environment
MPI_Finalize();
return 0;
}
```

**Walchand College of Engineering, Sangli**  
**Department of Computer Science and Engineering**

**Problem Statement 2:**

Implement a program to display rank and communicator group of five processes

**Screenshot #:**



The screenshot shows a Windows command prompt window titled "C:\Windows\System32\cmd.exe". The window displays the output of an MPI program executed with 10 processes. The output lists the rank and group for each process, all belonging to group -2013265920. The ranks are 6, 2, 7, 9, 3, 8, 4, 0, 5, and 1. The prompt is at the directory G:\Kunal\Sem-7\HPC Assignments\Assignment - 5.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2075]
(c) Microsoft Corporation. All rights reserved.

G:\Kunal\Sem-7\HPC Assignments\Assignment - 5>mpiexec.exe -np 10 Assignment5_2.exe
Rank: 6, Group: -2013265920
Rank: 2, Group: -2013265920
Rank: 7, Group: -2013265920
Rank: 9, Group: -2013265920
Rank: 3, Group: -2013265920
Rank: 8, Group: -2013265920
Rank: 4, Group: -2013265920
Rank: 0, Group: -2013265920
Rank: 5, Group: -2013265920
Rank: 1, Group: -2013265920

G:\Kunal\Sem-7\HPC Assignments\Assignment - 5>_
```

**Information #:**

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

/* run this program using the console pauser or add your own getch,
system("pause") or input loop */

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

    //Initialize the MPI environment
    MPI_Init(NULL,NULL);
```

**Walchand College of Engineering, Sangli**  
**Department of Computer Science and Engineering**

```
//Get the rank of process
int rank;
MPI_Comm_rank(MPI_COMM_WORLD, &rank);

MPI_Group group;
MPI_Comm_group(MPI_COMM_WORLD, &group);

printf("Rank: %d, Group: %d \n", rank, group);

//Finalize the MPI environment
MPI_Finalize();
return 0;
}
```

**Github Link:**

<https://github.com/Kunalkadam179/HPC-Assignment/tree/main/Assignment%20-%205>