Name: Varun Menon **Course Code: CSE4001**

Reg No: 19BCE1438 **Faculty: Dr. Harini S**

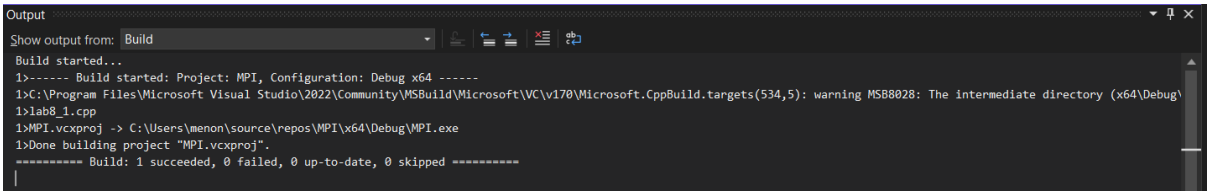
Lab Experiment 8

1. Sample Hello World

Code:

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

int main()
{
    printf("Hello World\n");
    return 0;
}
```

Output:



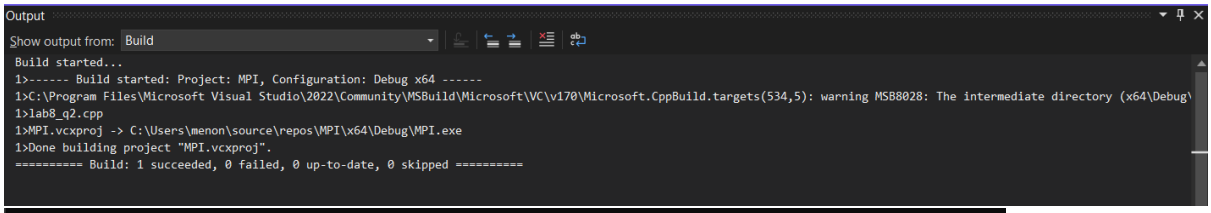

```
C:\Users\menon\source\repos\MPI\x64\Debug>mpiexec -n 6 ./MPI.exe
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

2. Prink rank, world size and processor name

Code:

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

int main(int argc, char **argv)
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);
    printf("World Size: %d\n", world_size);
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    printf("Processor Name: %s\n Rank: %d\n World Size: %d \n",
processor_name, world_rank, world_size);

    MPI_Finalize();
}
```

Output:



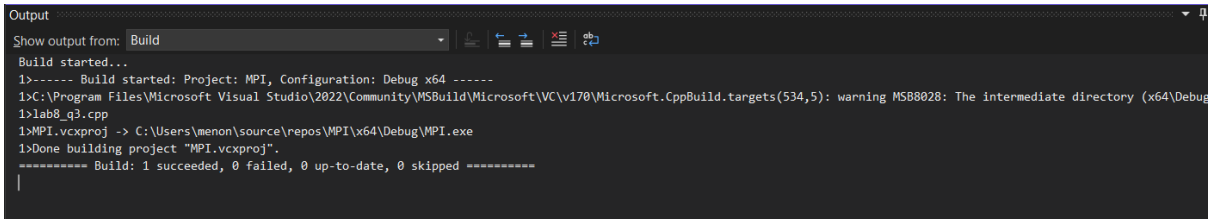

```
C:\Users\menon\source\repos\MPI\x64\Debug>mpiexec -n 3 ./MPI.exe
World Size: 3
Processor Name: LAPTOP-GEKLLQ23
 Rank: 2
 World Size: 3
World Size: 3
Processor Name: LAPTOP-GEKLLQ23
 Rank: 1
 World Size: 3
World Size: 3
Processor Name: LAPTOP-GEKLLQ23
 Rank: 0
 World Size: 3
```

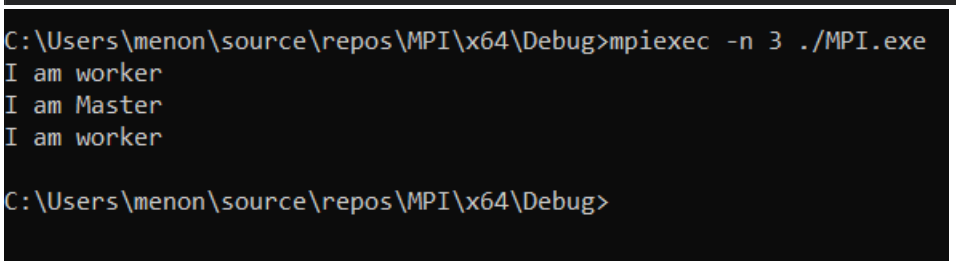
3. Master prints "I am Master", Worker prints "I am worker"

Code:

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

int main()
{
    int n = 3;
    int a[3][3] = {
        {1,2,3},
        {4,5,6},
        {7,8,9}
    };

    for(int i=1;i<n;i++)
    {
        #pragma omp parallel for
        for(int j=0;j<n;j++)
        {
            a[i][j] = a[i-1][j] + 2;
        }
    }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("a[%d][%d] = %d\n", i, j, a[i][j]);
        }
    }
}
```

Output:


Output
Show output from: Build
Build started...
1>------ Build started: Project: MPI, Configuration: Debug x64 ------
1>C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Microsoft\VC\v170\Microsoft.CppBuild.targets(534,5): warning MSB8028: The intermediate directory (x64\Debug
1>lab8_q3.cpp
1>MPI.vcxproj -> C:\Users\menon\source\repos\MPI\x64\Debug\MPI.exe
1>Done building project "MPI.vcxproj".
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========


C:\Users\menon\source\repos\MPI\x64\Debug>mpiexec -n 3 ./MPI.exe
I am worker
I am Master
I am worker
C:\Users\menon\source\repos\MPI\x64\Debug>

4. Master generates 1/2,1/4,1/8,1/16...1/n; Worker generates 2,4,8,16...n

Code:

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

int rank, numprocs, left, right, n;
MPI_Request request, request2;
MPI_Status status;

void slave_method()
{
    double buffer2[10];
    buffer2[0] = 2;
    for (int i = 1; i < n; i++)
    {
        buffer2[i] = buffer2[i - 1] * 2;
    }
    MPI_Isend(buffer2, n, MPI_DOUBLE, right, 123, MPI_COMM_WORLD,
&request2);
    MPI_Wait(&request2, &status);
    printf("By slave: ");
    for (int i = 0; i < n; i++)
    {
        printf("%f ", buffer2[i]);
    }
    printf("\n");
}

int main(int argc, char* argv[])
{
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    right = (rank + 1) % numprocs;
    left = rank - 1;
    if (left < 0)
        left = numprocs - 1;
    n = 10;
    double buffer[10];
    slave_method();
    MPI_Irecv(buffer, n, MPI_DOUBLE, left, 123, MPI_COMM_WORLD,
&request);
    MPI_Wait(&request, &status);
    MPI_Wait(&request2, &status);
    MPI_Finalize();
```

```
    if (rank == 0)
    {
        printf("From slave: ");
        for (int i = 0; i < n; i++)
        {
            printf("%f ", buffer[i]);
        }
        printf("\n");
        printf("By master: ");
        for (int i = 0; i < n; i++)
        {
            double value = 1.0 / buffer[i];
            printf("%f ", value);
        }
        printf("\n");
    }
    return 0;
}
```

Output: