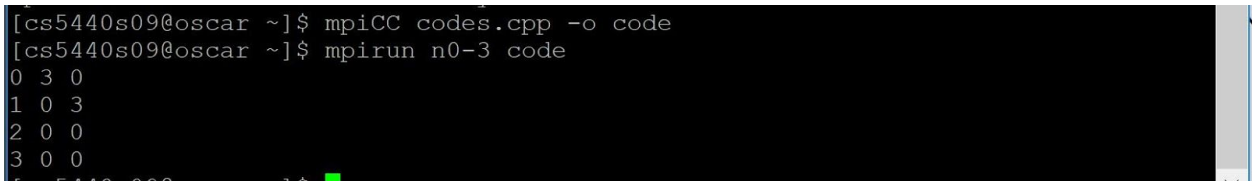
Question 13

10 lines of code will be generated

Question 14

For node0 a =3, b = 0; node1 a =0, b =3; node1 and 2 are 0 for a and b



Question 15

Code

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int sum;
MPI_Status status;
if(node == 0){
 MPI_Recv(&sum, 1, MPI_INT, node+1, 0, MPI_COMM_WORLD, &status);
sum = sum + 1 + 2;
 cout<<node<<" "<<sum<<endl;
}
else
 if(node == 1)
 {
sum = 3 + 4;
 MPI_Send(&sum, 1, MPI_INT, node-1, 0, MPI_COMM_WORLD);
 }

 MPI_Finalize();
return 0;
```

Output

```
[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun n0-1 code
0 10
```

output is 10.

Question 16

Code

/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP

Encoding Color

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int value;
MPI_Status status;
if(node == 9) {
int pi = 777;
for(int i = 0; i < 10; i++)
MPI_Send(&pi, 1, MPI_INT, i, 0, MPI_COMM_WORLD);
}

MPI_Recv(&value, 1, MPI_INT, 9, 0, MPI_COMM_WORLD, &status);
cout<<node<<" "<< value<<endl;


 MPI_Finalize();
return 0;
}
```

Output

```
[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun N code
0 777
3 777
2 777
4 777
1 777
6 777
5 777
7 777
8 777
9 777
```

Question 17

Code

/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP

Encoding Color

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int value;
MPI_Status status;

MPI_Send(&node, 1, MPI_INT, 9, 0, MPI_COMM_WORLD);

if(node == 9)
 {
int sum;
for(int i = 0; i < 10; i++){
MPI_Recv(&sum, 1, MPI_INT, i, 0, MPI_COMM_WORLD, &status);
if(i == 0) value = sum;
value = value + sum;
}
cout<<node<<" "<< value<<endl;
 }

 MPI_Finalize();
return 0;
}
```

Output

```
[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun N code
9 45
```

Question 18

```
[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun n0-3 code
5
0 5
3 5
1 5
2 5
```

Question 19

It broadcasts only to node0
output is;

```
[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun n0-3 code
1 0
0 3
3 0
2 0
```

Question 20

Nothing is displayed. None of the nodes receive any data because, all nodes keep waiting to receive and none of the nodes send.

Question 20

i) Print order in any form.

Code

/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP

Encoding Color

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Status status;

cout<<"Hello World"<<" "<<node<<" "<<endl;


 MPI_Finalize();
return 0;
}
```

Output

```
[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun N code
Hello World 0
Hello World 1
Hello World 5
Hello World 3
Hello World 4
Hello World 2
Hello World 8
Hello World 9
Hello World 6
Hello World 7
```

ii) Print order in form

Code

/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP

Encoding Color

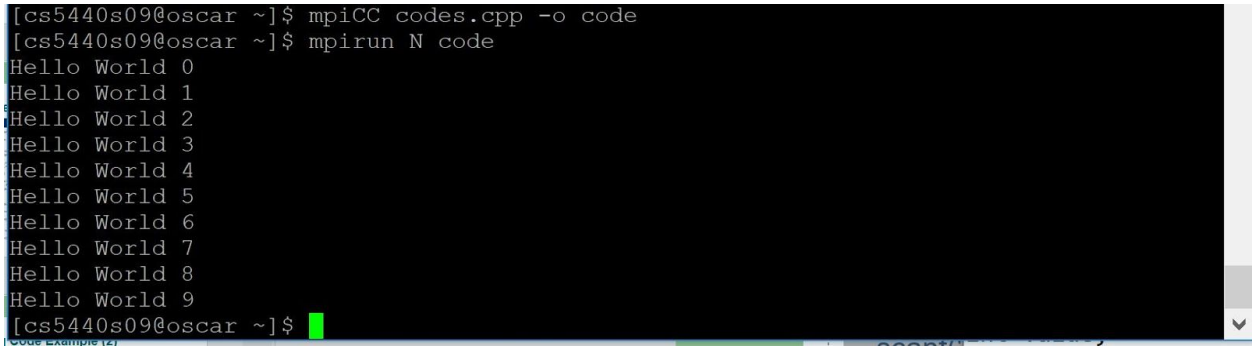
```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Status status;
int value;
if(node == 0)
  {
    MPI_Send(&node, 1, MPI_INT, node+1, 0, MPI_COMM_WORLD);
  }
  else{
    MPI_Recv(&value, 1, MPI_INT, node-1, 0, MPI_COMM_WORLD, &status);

    if(node < size - 1)
          MPI_Send(&value, 1, MPI_INT, node+1, 0, MPI_COMM_WORLD);

  }
cout<<"Hello World"<<" "<<node<<" "<<endl;

 MPI_Finalize();
return 0;
}
```

Output


[cs5440s09@oscar ~]$ mpiCC codes.cpp -o code
[cs5440s09@oscar ~]$ mpirun N code
Hello World 0
Hello World 1
Hello World 2
Hello World 3
Hello World 4
Hello World 5
Hello World 6
Hello World 7
Hello World 8
Hello World 9
[cs5440s09@oscar ~]$