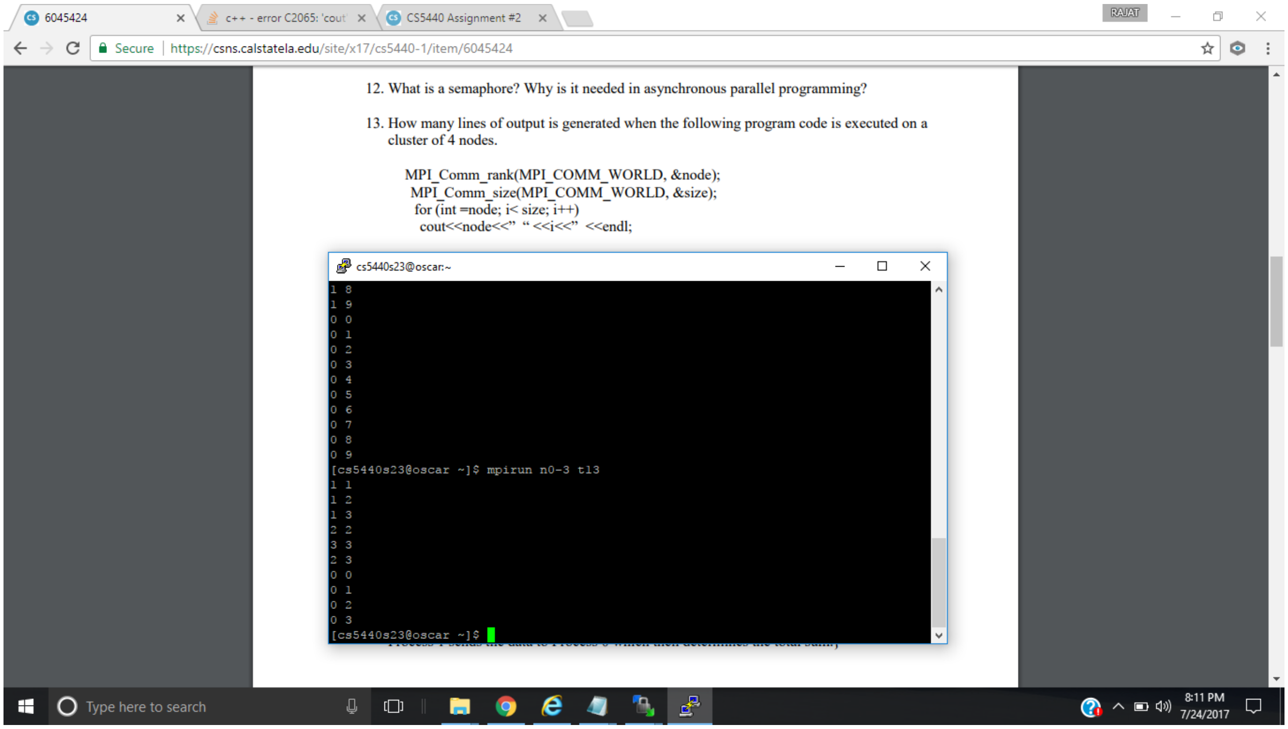
Assignment 2

Question 13



12. What is a semaphore? Why is it needed in asynchronous parallel programming?

13. How many lines of output is generated when the following program code is executed on a cluster of 4 nodes.

```
MPI_Comm_rank(MPI_COMM_WORLD, &node);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
  for (int =node; i< size; i++)
   cout<<node<<" " <<i<<" <<endl;
```

```
1 8
1 9
0 0
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0 9
[cs5440s23@oscar ~]$ mpirun n0-3 t13
1 1
1 2
1 3
2 2
3 3
2 3
0 0
0 1
0 2
0 3
[cs5440s23@oscar ~]$
```

Code:

```
#include <mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv)
{
int size, node,i;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
for(i=node; i< size; i++){
cout<<node<<" "<<i<<" "<<endl;
}
MPI_Finalize();
return 0;
}
```

Hand Trace:

Q13

Here Size = 4
For Node 0
for(i=0 ; i<4 ; i++)

```
0   1
0   2
0   3
```

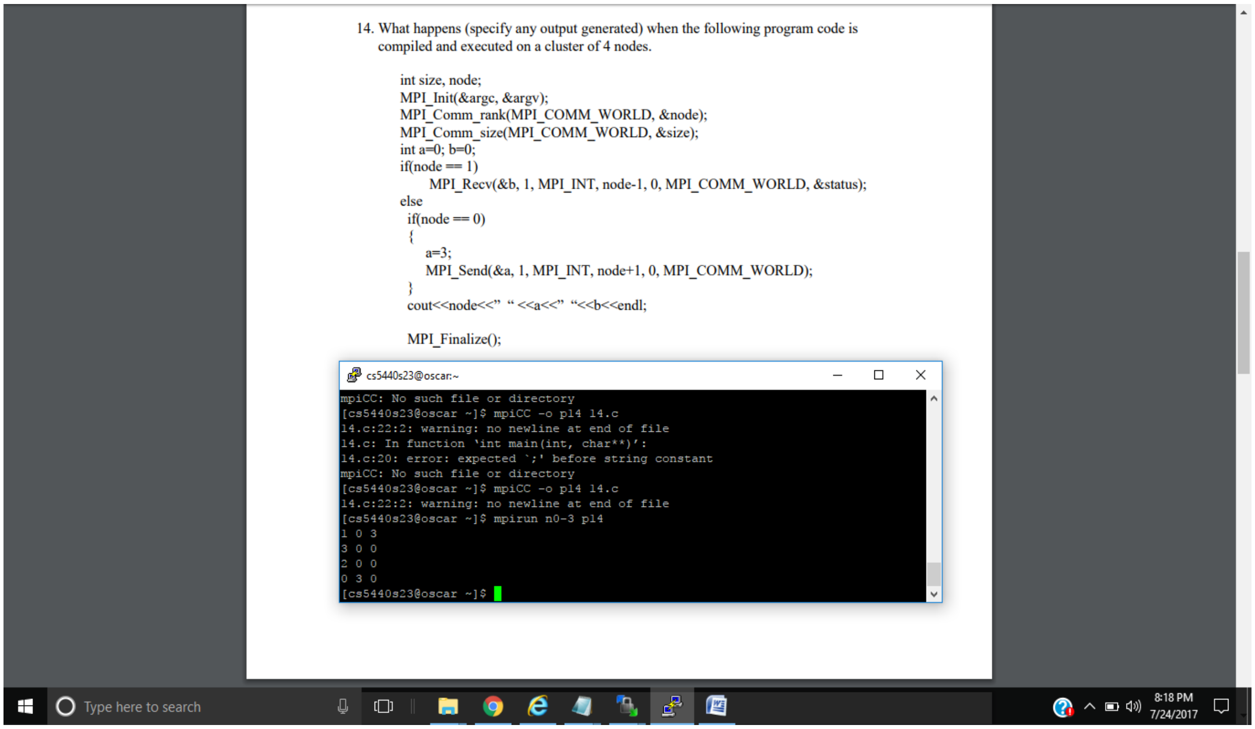
For Node 1
for (i=1 ; i<4 ; i+r)

```
1   1
1   2
1   3
```

There the final o/p will be

```
0   1
0   2
0   3
1   1
1   2
1   3
2   1
2   2
2   3
3   1
3   2
3   3
```

Question 14:



14. What happens (specify any output generated) when the following program code is compiled and executed on a cluster of 4 nodes.

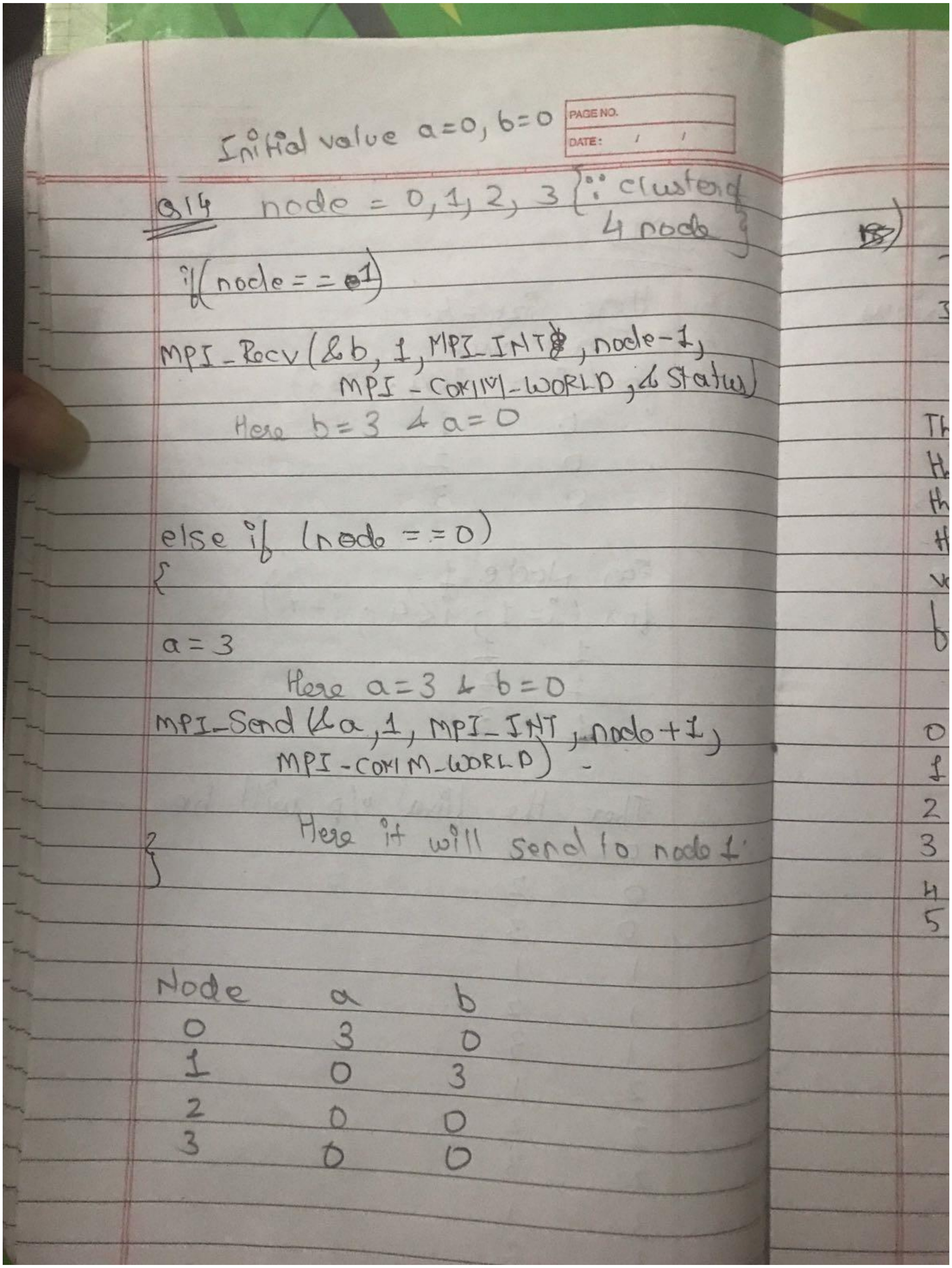
```
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int a=0; b=0;
if(node == 1)
    MPI_Recv(&b, 1, MPI_INT, node-1, 0, MPI_COMM_WORLD, &status);
else
 if(node == 0)
 {
   a=3;
   MPI_Send(&a, 1, MPI_INT, node+1, 0, MPI_COMM_WORLD);
 }
 cout<<node<<” “ <<a<<” “<<b<<endl;

 MPI_Finalize();
```

```
mpiCC: No such file or directory
[cs5440s23@oscar ~]$ mpiCC -o p14 14.c
14.c:22:2: warning: no newline at end of file
14.c: In function ‘int main(int, char**)’:
14.c:20: error: expected `;' before string constant
mpiCC: No such file or directory
[cs5440s23@oscar ~]$ mpiCC -o p14 14.c
14.c:22:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun n0-3 p14
1 0 3
3 0 0
2 0 0
0 3 0
[cs5440s23@oscar ~]$
```

Code:

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

Hand Trace:



Initial value a=0, b=0

PAGE NO.

DATE: / /

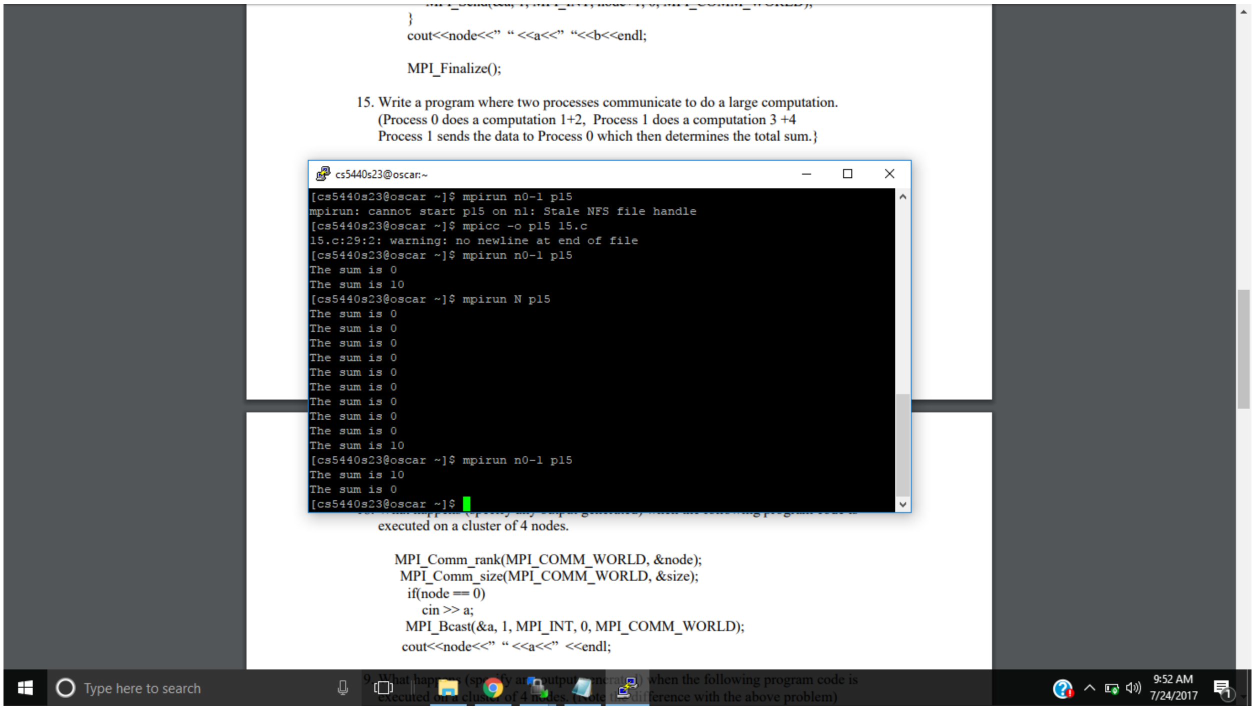
Q14 node = 0, 1, 2, 3 {∵ clustered 4 node }

if(node == 1)

```
MPI_Recv(&b, 1, MPI_INT, node-1,
        MPI_COMM_WORLD, & status)
    Here b=3 & a=0

else if (node == 0)
{

a = 3
        Here a=3 & b=0
MPI_Send(&a, 1, MPI_INT, node+1,
        MPI_COMM_WORLD)

        Here it will send to node 1
}
```

Node	a	b
0	3	0
1	0	3
2	0	0
3	0	0

Question 15:



```
}
cout<<node<<" " <<a<<" "<<b<<endl;

MPI_Finalize();
```

15. Write a program where two processes communicate to do a large computation.
(Process 0 does a computation 1+2, Process 1 does a computation 3 +4
Process 1 sends the data to Process 0 which then determines the total sum.}

```
[cs5440s23@oscar ~]$ mpirun n0-1 p15
mpirun: cannot start p15 on n1: Stale NFS file handle
[cs5440s23@oscar ~]$ mpicc -o p15 15.c
15.c:29:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun n0-1 p15
The sum is 0
The sum is 10
[cs5440s23@oscar ~]$ mpirun N p15
The sum is 0
The sum is 0
The sum is 0
The sum is 0
The sum is 0
The sum is 0
The sum is 0
The sum is 0
The sum is 0
The sum is 10
[cs5440s23@oscar ~]$ mpirun n0-1 p15
The sum is 10
The sum is 0
[cs5440s23@oscar ~]$
```

executed on a cluster of 4 nodes.

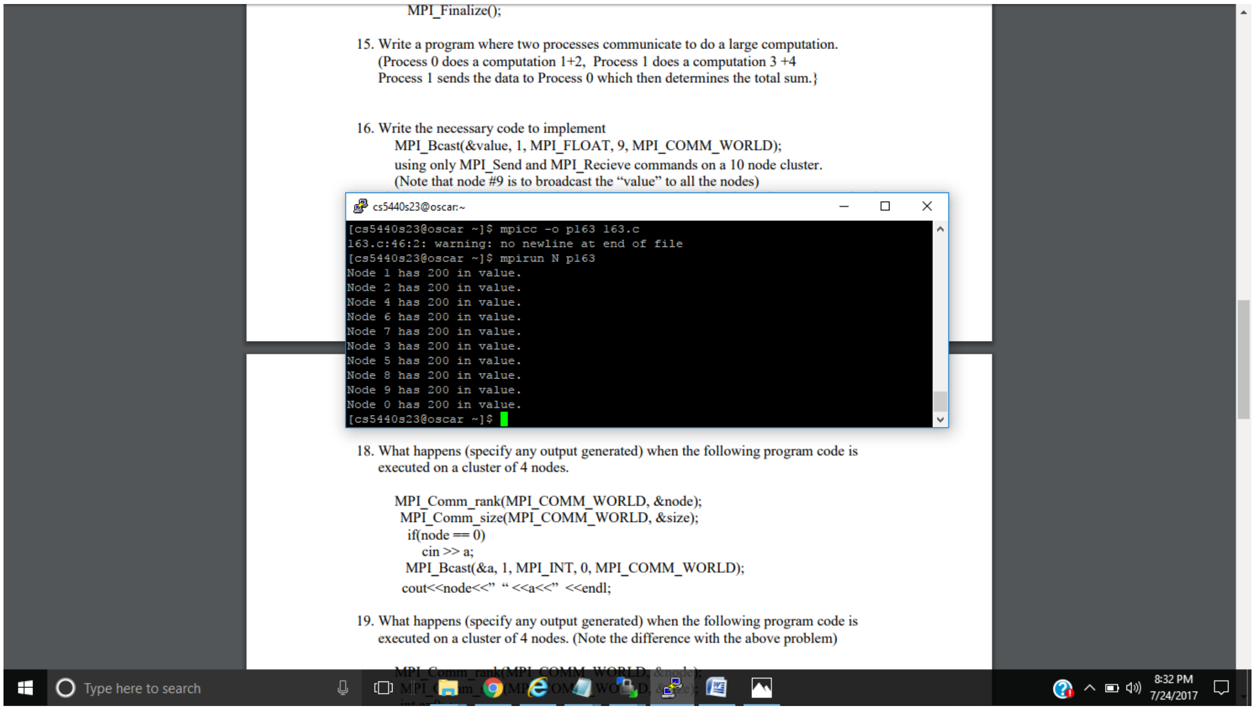
```
MPI_Comm_rank(MPI_COMM_WORLD, &node);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
  if(node == 0)
     cin >> a;
 MPI_Bcast(&a, 1, MPI_INT, 0, MPI_COMM_WORLD);
cout<<node<<" " <<a<<" <<endl;
```

9. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes. (Note the difference with the above problem)

Code:

```
#include<stdio.h>
#include <mpi.h>
int main(int argc, char** argv)
{
int size, node;
MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int a=0, b=0, sum=0;
if(node == 1){
MPI_Recv(&a, 1, MPI_INT, 0, 0, MPI_COMM_WORLD,&status);
b=3+4;
MPI_Send(&b, 1, MPI_INT, 0, 0, MPI_COMM_WORLD);

}else
if(node == 0)
{
a=1+2;
MPI_Send(&a, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);

MPI_Recv(&b, 1, MPI_INT, 1, 0, MPI_COMM_WORLD,&status);

sum= a+b;
}
printf("The sum is %d \n",sum);
MPI_Finalize();
}
```

Question 16:



MPI_Finalize();

15. Write a program where two processes communicate to do a large computation. (Process 0 does a computation 1+2, Process 1 does a computation 3 +4 Process 1 sends the data to Process 0 which then determines the total sum.}

16. Write the necessary code to implement
    MPI_Bcast(&value, 1, MPI_FLOAT, 9, MPI_COMM_WORLD);
    using only MPI_Send and MPI_Recieve commands on a 10 node cluster.
    (Note that node #9 is to broadcast the "value" to all the nodes)

```
[cs5440s23@oscar ~]$ mpicc -o p163 163.c
163.c:46:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p163
Node 1 has 200 in value.
Node 2 has 200 in value.
Node 4 has 200 in value.
Node 6 has 200 in value.
Node 7 has 200 in value.
Node 3 has 200 in value.
Node 5 has 200 in value.
Node 8 has 200 in value.
Node 9 has 200 in value.
Node 0 has 200 in value.
[cs5440s23@oscar ~]$
```

18. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes.

```
MPI_Comm_rank(MPI_COMM_WORLD, &node);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
  if(node == 0)
     cin >> a;
 MPI_Bcast(&a, 1, MPI_INT, 0, MPI_COMM_WORLD);
cout<<node<<" " <<a<<" <<endl;
```

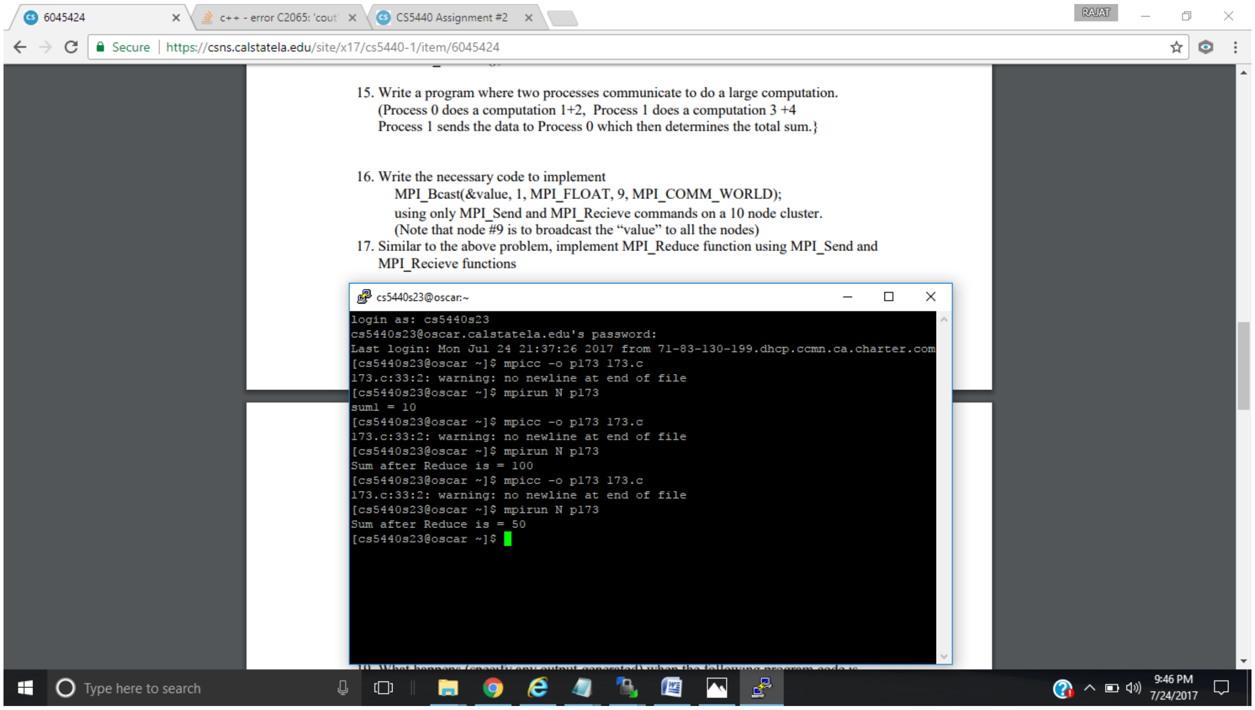
19. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes. (Note the difference with the above problem)

```
MPI_Comm_rank(MPI_COMM_WORLD, &node);
```

Code:

```
#include <mpi.h>
#include<stdio.h>
int main(int argc, char** argv)
{
int size, node,x;
MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int a=0, b=0, value=100;
float sum=0;
if(node==9)
{
value=200;
for(x=0;x<=8;x++){
MPI_Send(&value, 1, MPI_INT, x, 0, MPI_COMM_WORLD);
}
}
MPI_Recv(&value, 1, MPI_INT, 9, 0, MPI_COMM_WORLD,&status);
printf("Node %d has %d in value. \n", node,value);
MPI_Finalize();
}
```

Question 17:



15. Write a program where two processes communicate to do a large computation.
(Process 0 does a computation 1+2, Process 1 does a computation 3 +4
Process 1 sends the data to Process 0 which then determines the total sum.}

16. Write the necessary code to implement
MPI_Bcast(&value, 1, MPI_FLOAT, 9, MPI_COMM_WORLD);
using only MPI_Send and MPI_Recieve commands on a 10 node cluster.
(Note that node #9 is to broadcast the "value" to all the nodes)

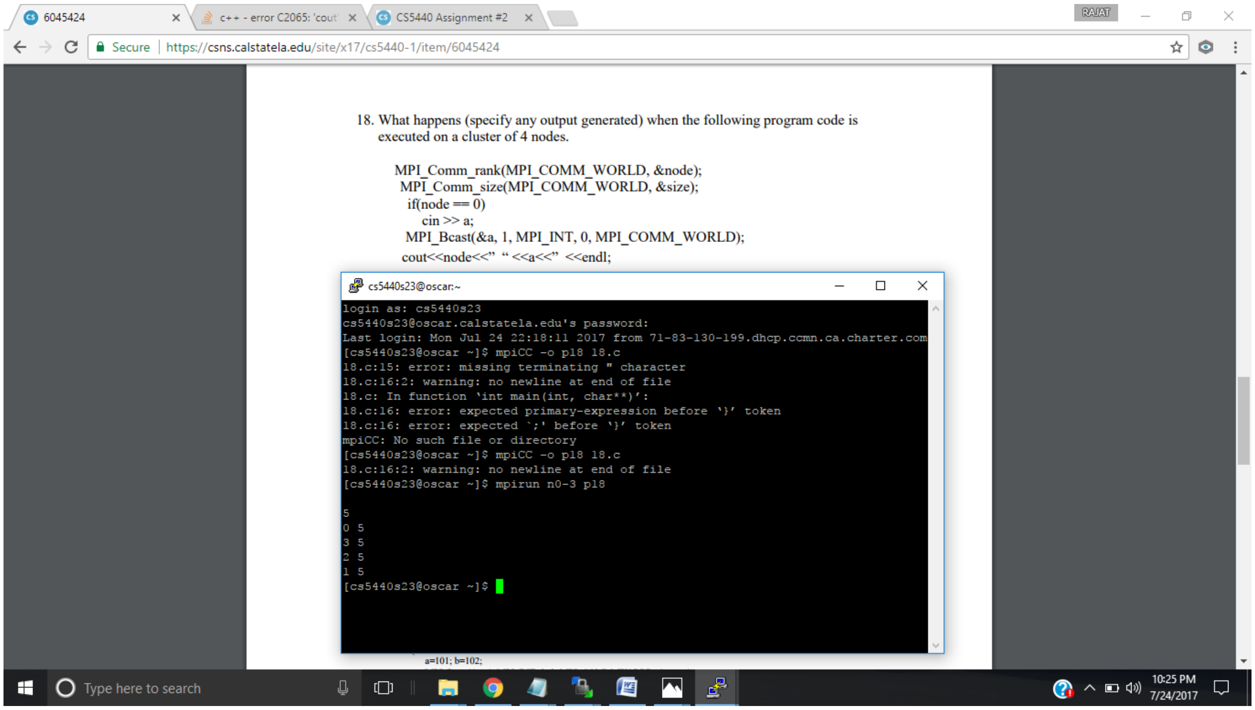
17. Similar to the above problem, implement MPI_Reduce function using MPI_Send and MPI_Recieve functions

```
login as: cs5440s23
cs5440s23@oscar.calstatela.edu's password:
Last login: Mon Jul 24 21:37:26 2017 from 71-83-130-199.dhcp.ccmn.ca.charter.com
[cs5440s23@oscar ~]$ mpicc -o p173 173.c
173.c:33:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p173
sum1 = 10
[cs5440s23@oscar ~]$ mpicc -o p173 173.c
173.c:33:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p173
Sum after Reduce is = 100
[cs5440s23@oscar ~]$ mpicc -o p173 173.c
173.c:33:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p173
Sum after Reduce is = 50
[cs5440s23@oscar ~]$
```

Code

```
#include <mpi.h>
#include<stdio.h>
int main(int argc, char** argv)
{
int size, node,x;
MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int a=0, b=0,s=98, sum=0, value=10;
if(node==9)
{
for(x=0;x<=9;x++){
MPI_Send(&value, 1, MPI_INT, x, 0, MPI_COMM_WORLD);
}
}
MPI_Recv(&value, 1, MPI_INT, 9, 0, MPI_COMM_WORLD,&status);
for(x=0;x<=9;x++)
{
sum=sum+value;
}
if(node==0)
        printf("sum1 = %d \n",sum);

MPI_Finalize();
}
```

Question 18:



6045424
c++ - error C2065: 'cout'
CS5440 Assignment #2
RAJAT
Secure
https://csns.calstatela.edu/site/x17/cs5440-1/item/6045424

18. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes.

```
MPI_Comm_rank(MPI_COMM_WORLD, &node);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
  if(node == 0)
      cin >> a;
 MPI_Bcast(&a, 1, MPI_INT, 0, MPI_COMM_WORLD);
cout<<node<<" " <<a<<" <<endl;
```

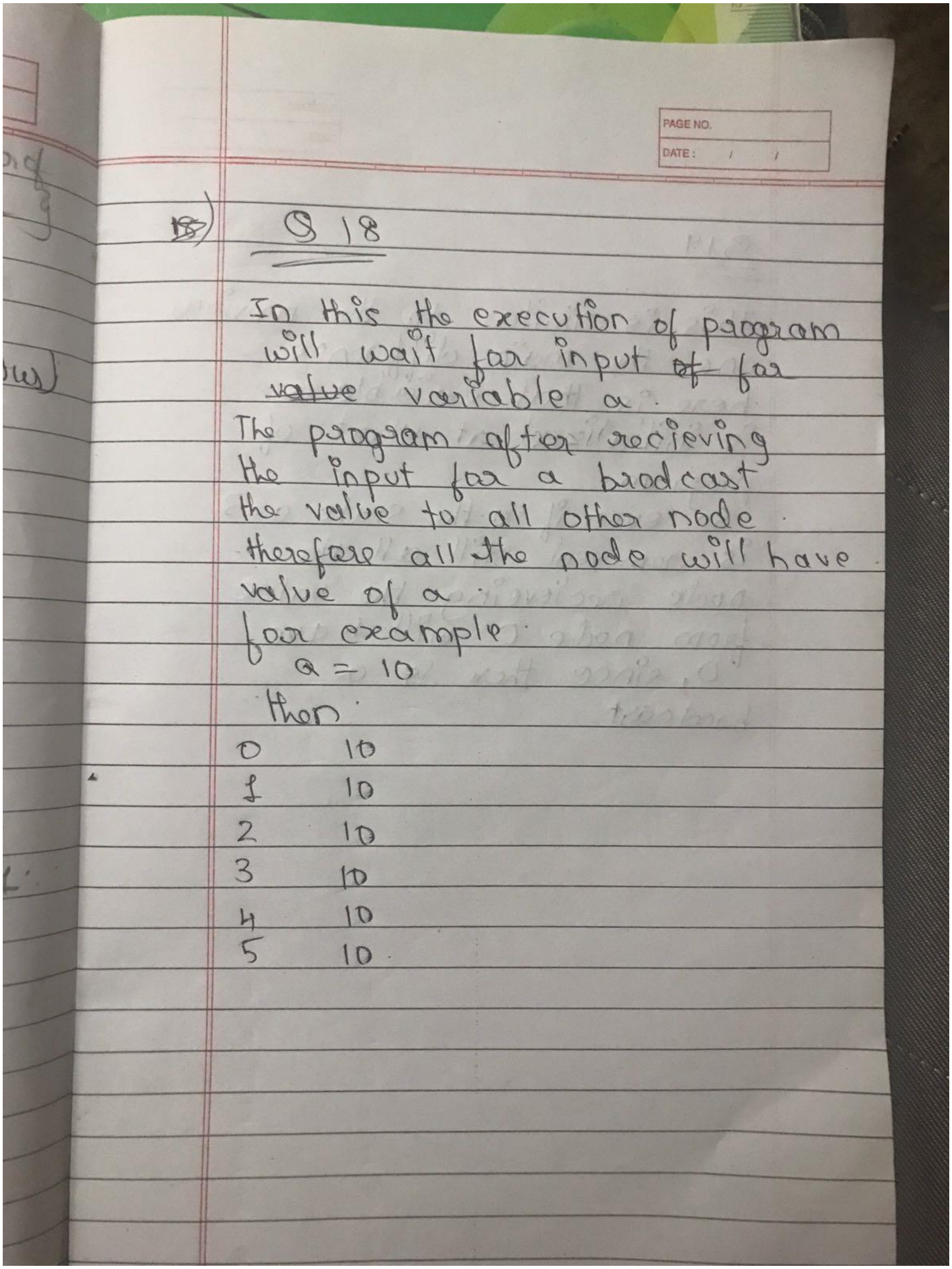
```
login as: cs5440s23
cs5440s23@oscar.calstatela.edu's password:
Last login: Mon Jul 24 22:18:11 2017 from 71-83-130-199.dhcp.ccmn.ca.charter.com
[cs5440s23@oscar ~]$ mpiCC -o p18 18.c
18.c:15: error: missing terminating " character
18.c:16:2: warning: no newline at end of file
18.c: In function 'int main(int, char**)':
18.c:16: error: expected primary-expression before '}' token
18.c:16: error: expected `;' before '}' token
mpiCC: No such file or directory
[cs5440s23@oscar ~]$ mpiCC -o p18 18.c
18.c:16:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun n0-3 p18

5
0 5
3 5
2 5
1 5
[cs5440s23@oscar ~]$
```

a=101; b=102;

<u>Code:</u>

```
#include <mpi.h>
#include<stdio.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv)
{
int size, node,a;
MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
 if(node == 0)
 cin >> a;
 MPI_Bcast(&a, 1, MPI_INT, 0, MPI_COMM_WORLD);
 cout<<node<<" " <<a<<" "<<endl;
}
```

Hand Trace:



PAGE NO.
DATE : / /

18) Q 18

In this the execution of program will wait for input ~~of~~ for ~~value~~ variable a.

The program after recieving the input for a brodcast the value to all other node. therefore all the node will have value of a.

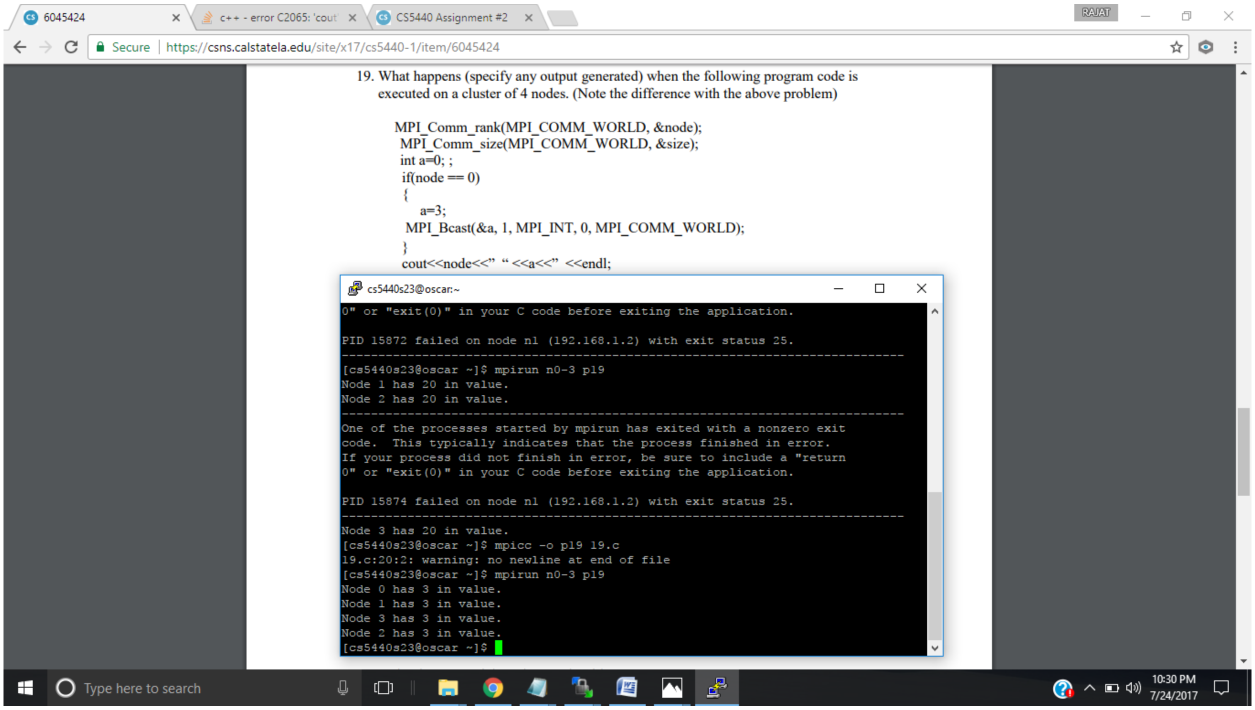
for example.

a = 10

then.

0	10
1	10
2	10
3	10
4	10
5	10.

Question 19:



19. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes. (Note the difference with the above problem)

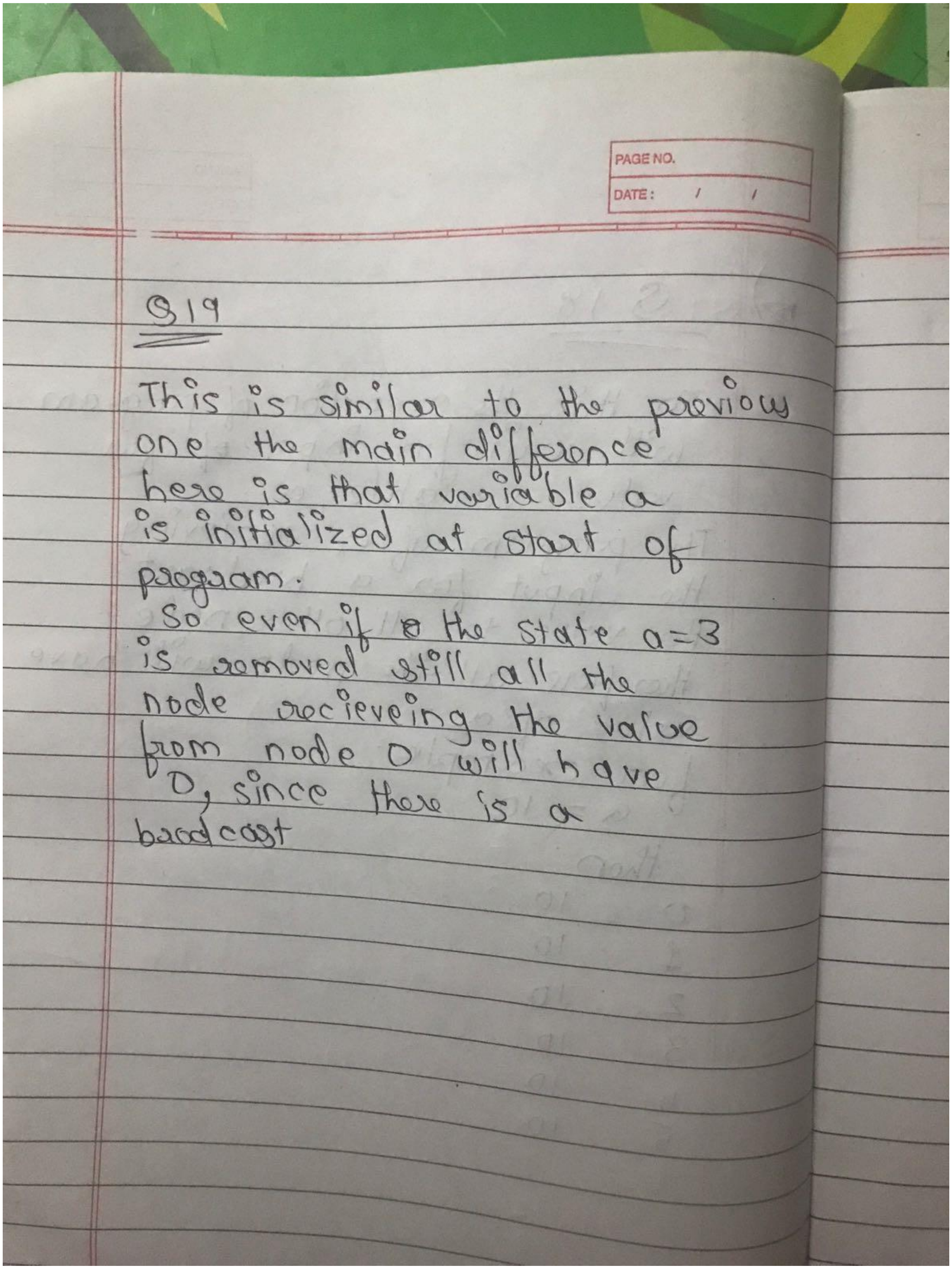
```
MPI_Comm_rank(MPI_COMM_WORLD, &node);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
 int a=0; ;
 if(node == 0)
 {
     a=3;
 MPI_Bcast(&a, 1, MPI_INT, 0, MPI_COMM_WORLD);
 }
 cout<<node<<" " <<a<<" <<endl;
```

```
0" or "exit(0)" in your C code before exiting the application.

PID 15872 failed on node n1 (192.168.1.2) with exit status 25.
-----------------------------------------------------------------------------
[cs5440s23@oscar ~]$ mpirun n0-3 p19
Node 1 has 20 in value.
Node 2 has 20 in value.
-----------------------------------------------------------------------------
One of the processes started by mpirun has exited with a nonzero exit
code.  This typically indicates that the process finished in error.
If your process did not finish in error, be sure to include a "return
0" or "exit(0)" in your C code before exiting the application.

PID 15874 failed on node n1 (192.168.1.2) with exit status 25.
-----------------------------------------------------------------------------
Node 3 has 20 in value.
[cs5440s23@oscar ~]$ mpicc -o p19 19.c
19.c:20:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun n0-3 p19
Node 0 has 3 in value.
Node 1 has 3 in value.
Node 3 has 3 in value.
Node 2 has 3 in value.
[cs5440s23@oscar ~]$
```

Code:

```
#include <mpi.h>
#include<stdio.h>
int main(int argc, char** argv)
{
int size, node;
MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int a=0;
 if(node == 0)
  a=3;
 MPI_Bcast(&a, 1, MPI_INT, 0, MPI_COMM_WORLD);
 printf("Node %d has %d in value. \n", node,a);
MPI_Finalize();
}
```

Hand Trace:



PAGE NO.

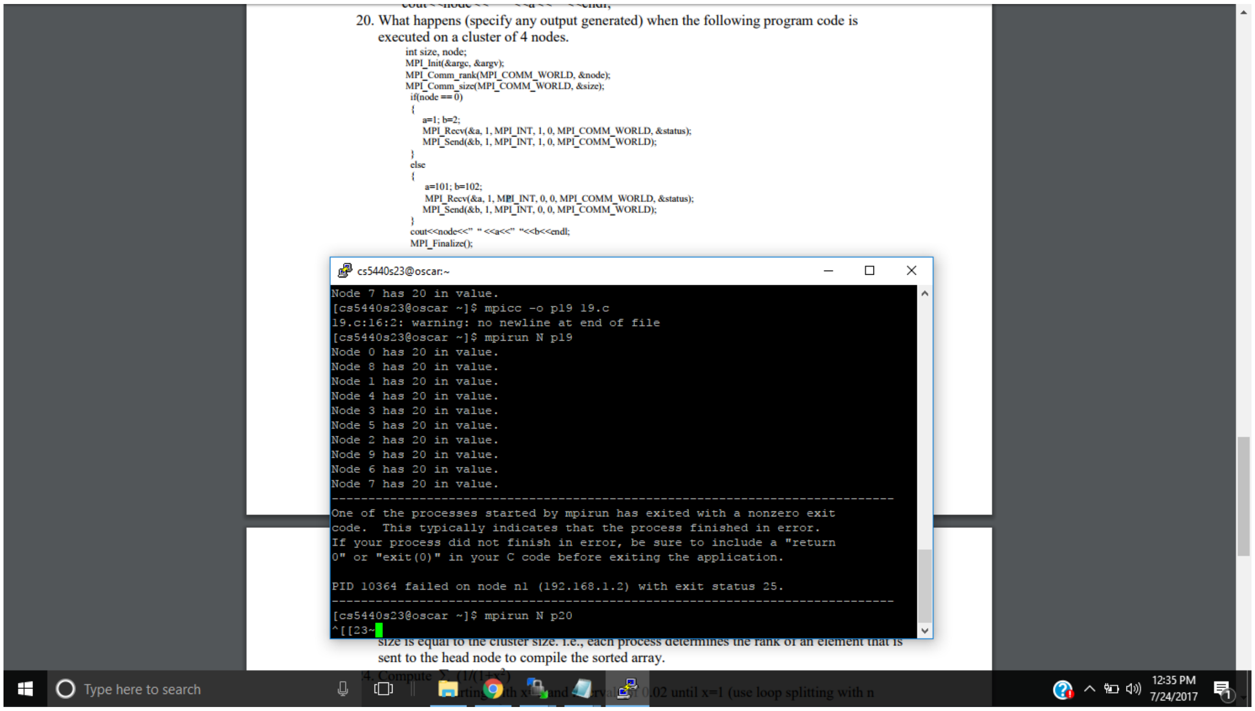
DATE : / /

Q19

This is similar to the previous one the main difference here is that variable a is initialized at start of program.

So even if the state a=3 is removed still all the node recieveing the value from node 0 will have 0, since there is a broadcast

Question 20:



20. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes.

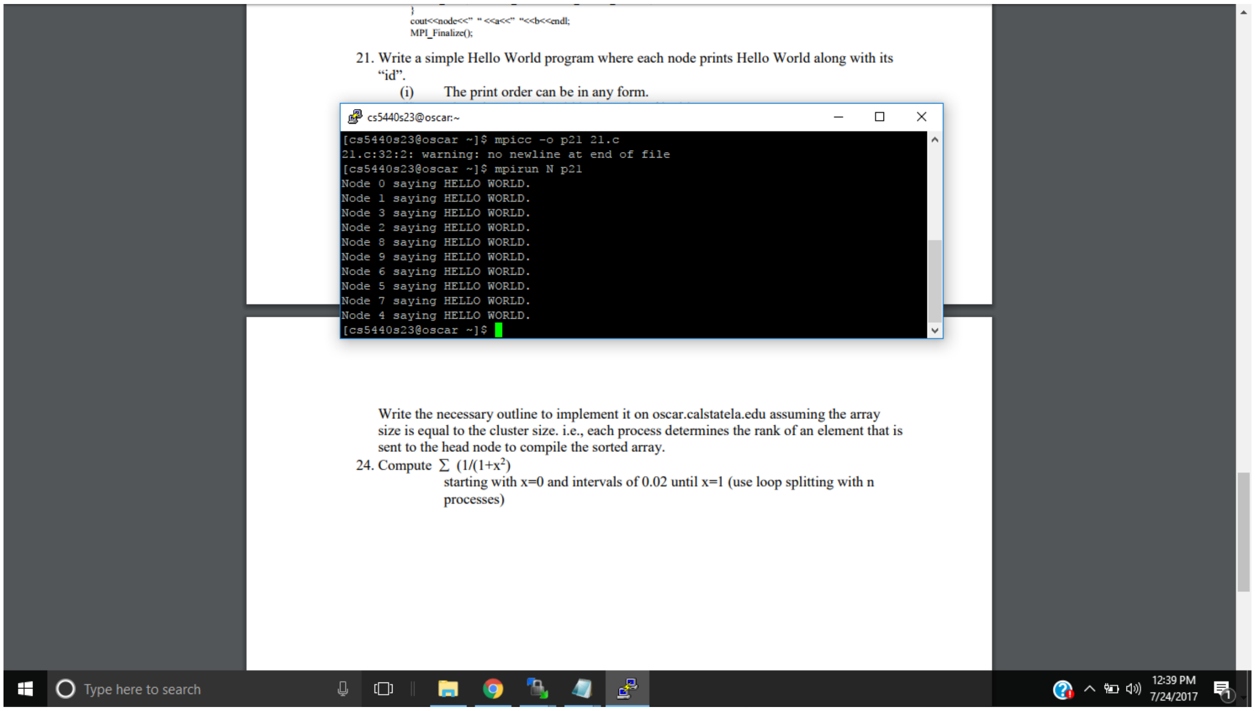
```
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
 if(node == 0)
 {
    a=1; b=2;
    MPI_Recv(&a, 1, MPI_INT, 1, 0, MPI_COMM_WORLD, &status);
    MPI_Send(&b, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
 }
 else
 {
     a=101; b=102;
     MPI_Recv(&a, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, &status);
    MPI_Send(&b, 1, MPI_INT, 0, 0, MPI_COMM_WORLD);
 }
 cout<<node<<" " <<a<<" "<<b<<endl;
 MPI_Finalize();
```

```
Node 7 has 20 in value.
[cs5440s23@oscar ~]$ mpicc -o p19 19.c
19.c:16:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p19
Node 0 has 20 in value.
Node 8 has 20 in value.
Node 1 has 20 in value.
Node 4 has 20 in value.
Node 3 has 20 in value.
Node 5 has 20 in value.
Node 2 has 20 in value.
Node 9 has 20 in value.
Node 6 has 20 in value.
Node 7 has 20 in value.
-----------------------------------------------------------------------------
One of the processes started by mpirun has exited with a nonzero exit
code.  This typically indicates that the process finished in error.
If your process did not finish in error, be sure to include a "return
0" or "exit(0)" in your C code before exiting the application.

PID 10364 failed on node n1 (192.168.1.2) with exit status 25.
-----------------------------------------------------------------------------
[cs5440s23@oscar ~]$ mpirun N p20
^[[23~
```

sent to the head node to compile the sorted array.

Both process goes in a state of indefinite wait where first process waiting for second to send data and second waiting for first to send data.

Question 21(i)



```
 }
 cout<<node<<" " <<a<<" "<<b<<endl;
 MPI_Finalize();
```

21. Write a simple Hello World program where each node prints Hello World along with its "id".
    (i) The print order can be in any form.

```
[cs5440s23@oscar ~]$ mpicc -o p21 21.c
21.c:32:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p21
Node 0 saying HELLO WORLD.
Node 1 saying HELLO WORLD.
Node 3 saying HELLO WORLD.
Node 2 saying HELLO WORLD.
Node 8 saying HELLO WORLD.
Node 9 saying HELLO WORLD.
Node 6 saying HELLO WORLD.
Node 5 saying HELLO WORLD.
Node 7 saying HELLO WORLD.
Node 4 saying HELLO WORLD.
[cs5440s23@oscar ~]$
```

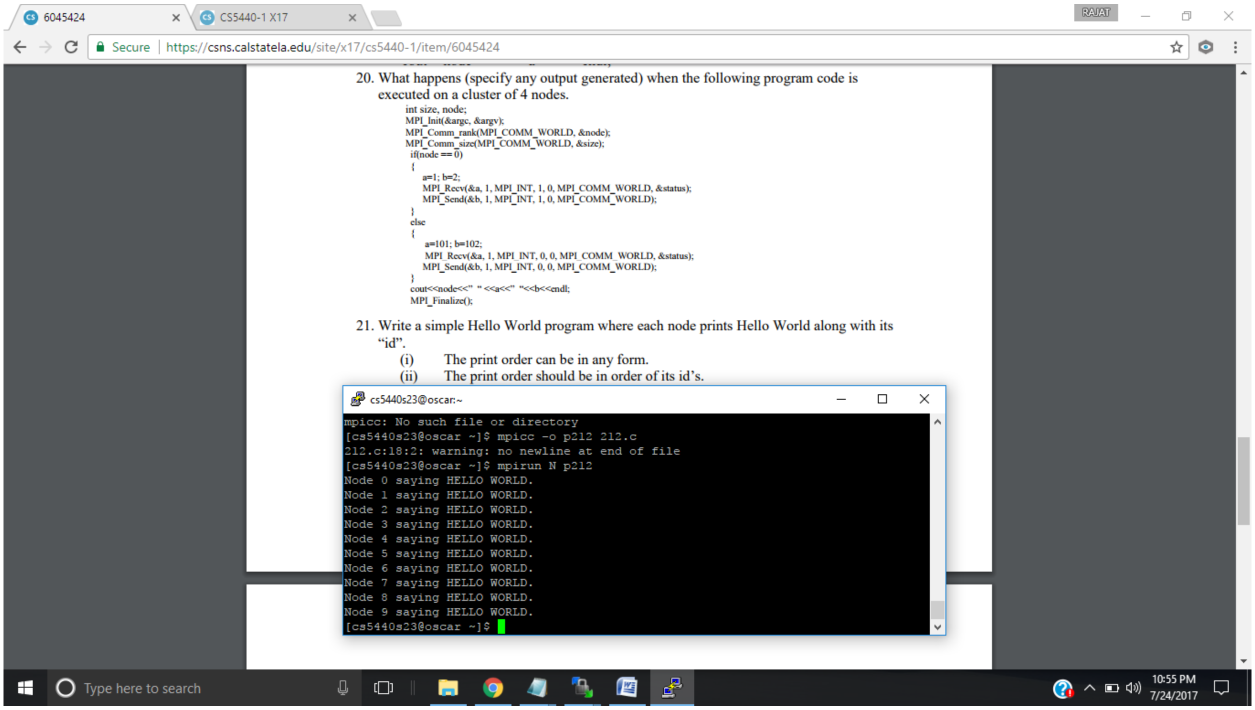
Write the necessary outline to implement it on oscar.calstatela.edu assuming the array size is equal to the cluster size. i.e., each process determines the rank of an element that is sent to the head node to compile the sorted array.

24. Compute Σ (1/(1+x^2))
    starting with x=0 and intervals of 0.02 until x=1 (use loop splitting with n processes)

Code:

```
#include <mpi.h>
#include<stdio.h>
int main(int argc, char** argv)
{
int size, node;
MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
printf("Node %d saying HELLO WORLD. \n", node);
MPI_Finalize();
}
```

Question 21(ii)



20. What happens (specify any output generated) when the following program code is executed on a cluster of 4 nodes.

```
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
 if(node == 0)
 {
    a=1; b=2;
    MPI_Recv(&a, 1, MPI_INT, 1, 0, MPI_COMM_WORLD, &status);
    MPI_Send(&b, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
 }
 else
 {
     a=101; b=102;
     MPI_Recv(&a, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, &status);
    MPI_Send(&b, 1, MPI_INT, 0, 0, MPI_COMM_WORLD);
 }
 cout<<node<<" " <<a<<" "<<b<<endl;
 MPI_Finalize();
```

21. Write a simple Hello World program where each node prints Hello World along with its "id".
    (i) The print order can be in any form.
    (ii) The print order should be in order of its id's.

```
mpicc: No such file or directory
[cs5440s23@oscar ~]$ mpicc -o p212 212.c
212.c:18:2: warning: no newline at end of file
[cs5440s23@oscar ~]$ mpirun N p212
Node 0 saying HELLO WORLD.
Node 1 saying HELLO WORLD.
Node 2 saying HELLO WORLD.
Node 3 saying HELLO WORLD.
Node 4 saying HELLO WORLD.
Node 5 saying HELLO WORLD.
Node 6 saying HELLO WORLD.
Node 7 saying HELLO WORLD.
Node 8 saying HELLO WORLD.
Node 9 saying HELLO WORLD.
[cs5440s23@oscar ~]$
```

Code:

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char** argv)
{
  int size, node;
 MPI_Status status;
 MPI_Init(&argc, &argv);
 MPI_Comm_rank(MPI_COMM_WORLD, &node);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
int value1=20, value2=22;
if(node>0)
```

```
  MPI_Recv(&value1, 1, MPI_INT, node-1, 0, MPI_COMM_WORLD, &status);
printf("Node %d saying HELLO WORLD. \n", node);
 if(node<9)
  MPI_Send(&value2, 1, MPI_INT, node+1, 0, MPI_COMM_WORLD);
 MPI_Finalize();
 return 0;
}
```