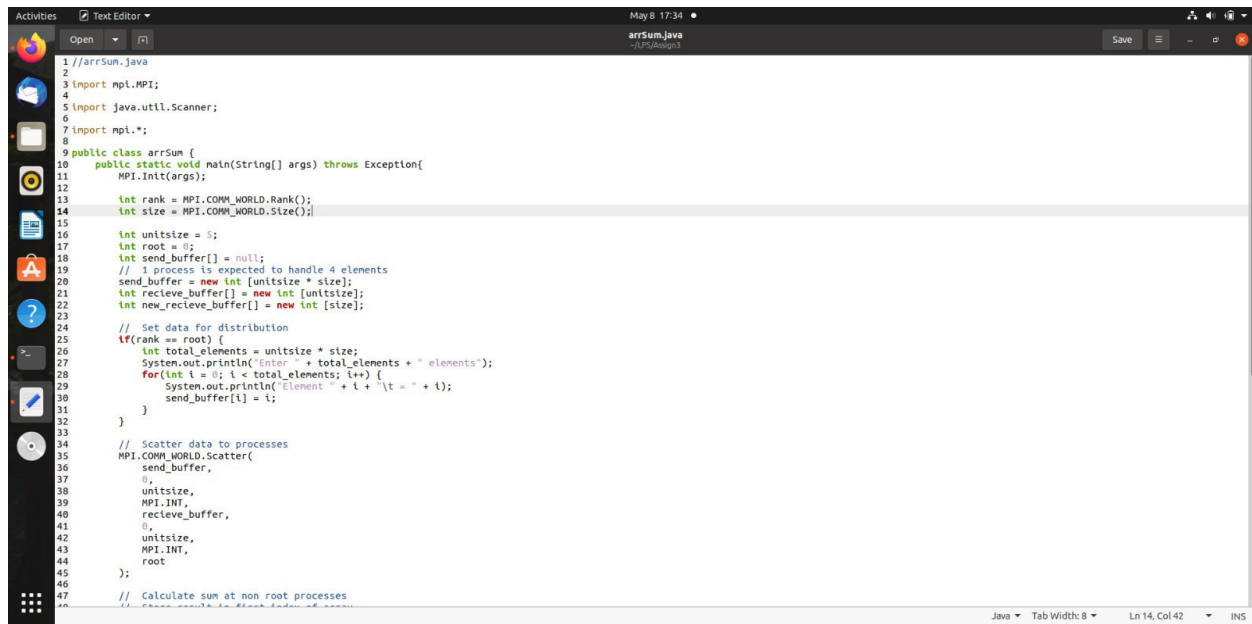


Name- Vaibhav Bichave

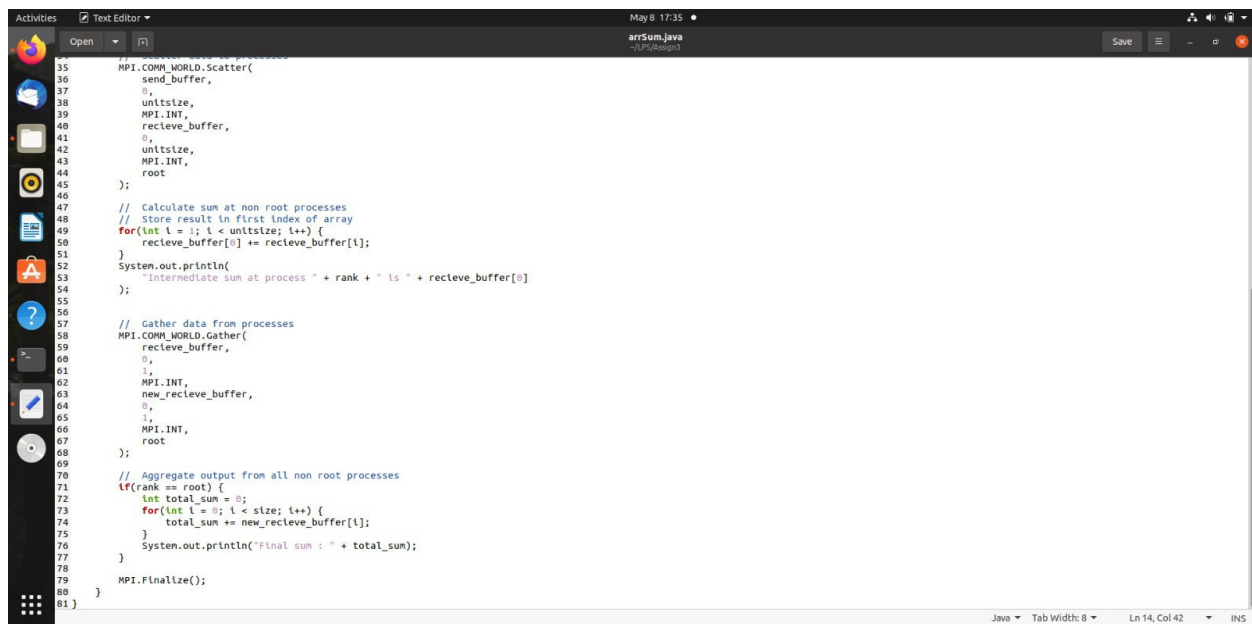
Roll no.-43209

## CODE IMPLEMENTATION :



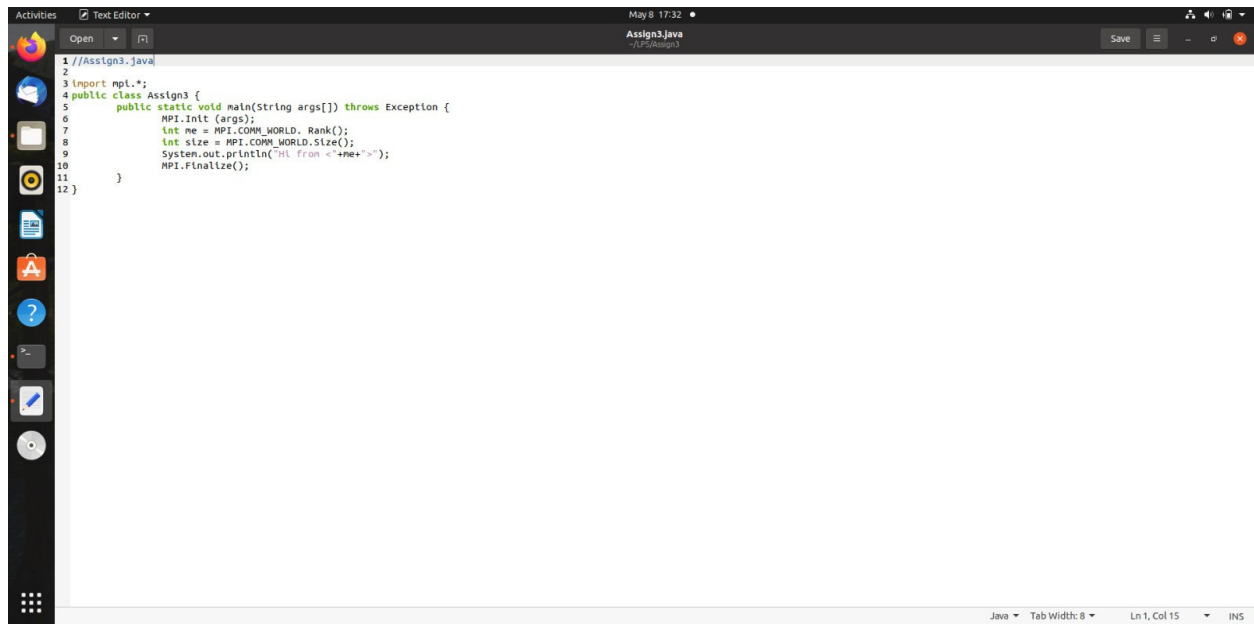
The screenshot shows a text editor window titled 'arrSum.java' with the following code:

```
1 //arrSum.java
2
3 import mpi.MPI;
4
5 import java.util.Scanner;
6
7 import mpi.*;
8
9 public class arrSum {
10     public static void main(String[] args) throws Exception{
11         MPI.Init(args);
12
13         int rank = MPI.COMM_WORLD.Rank();
14         int size = MPI.COMM_WORLD.Size();
15
16         int unitSize = 5;
17         int root = 0;
18         int send_buffer[] = null;
19         // 1 process is expected to handle 4 elements
20         send_buffer = new int [unitSize * size];
21         int receive_buffer[] = new int [unitSize];
22         int new_receive_buffer[] = new int [size];
23
24         // Set data for distribution
25         if(rank == root) {
26             int total_elements = unitSize * size;
27             System.out.println("Enter " + total_elements + " elements");
28             for(int i = 0; i < total_elements; i++) {
29                 System.out.println("Element " + i + ": ");
30                 send_buffer[i] = i;
31             }
32         }
33
34         // Scatter data to processes
35         MPI.COMM_WORLD.Scatter(
36             send_buffer,
37             0,
38             unitSize,
39             MPI.INT,
40             receive_buffer,
41             0,
42             unitSize,
43             MPI.INT,
44             root
45         );
46
47         // Calculate sum at non root processes
```



The screenshot shows the continuation of the code in the text editor window:

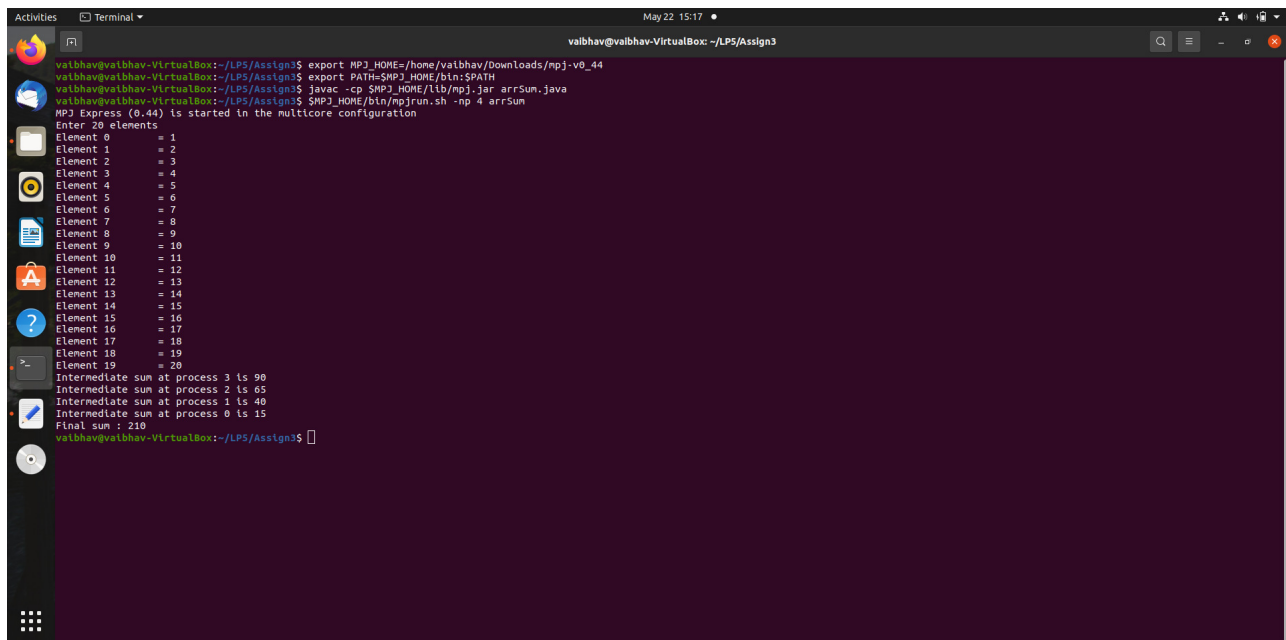
```
35         MPI.COMM_WORLD.Scatter(
36             send_buffer,
37             0,
38             unitSize,
39             MPI.INT,
40             receive_buffer,
41             0,
42             unitSize,
43             MPI.INT,
44             root
45         );
46
47         // Calculate sum at non root processes
48         // Store result in first index of array
49         for(int i = 0; i < unitSize; i++) {
50             receive_buffer[0] += receive_buffer[i];
51         }
52         System.out.println(
53             "Intermediate sum at process " + rank + " is " + receive_buffer[0]
54         );
55
56         // Gather data from processes
57         MPI.COMM_WORLD.Gather(
58             receive_buffer,
59             0,
60             1,
61             MPI.INT,
62             new_receive_buffer,
63             0,
64             1,
65             MPI.INT,
66             root
67         );
68
69         // Aggregate output from all non root processes
70         if(rank == root) {
71             int total_sum = 0;
72             for(int i = 0; i < size; i++) {
73                 total_sum += new_receive_buffer[i];
74             }
75             System.out.println("Final sum : " + total_sum);
76         }
77
78         MPI.Finalize();
79
80     }
81 }
```



The screenshot shows a Text Editor window titled "Assign3.java" with the following code:

```
1 //Assign3.java
2
3 import mpi.*;
4 public class Assign3 {
5     public static void main(String args[]) throws Exception {
6         MPI.Intt(args);
7         int ne = MPI.COMM_WORLD.Rank();
8         int size = MPI.COMM_WORLD.Size();
9         System.out.println("Hi from <"+ne+">");
10        MPI.Finalize();
11    }
12 }
```

The status bar at the bottom indicates "Java", "Tab Width: 8", "Ln 1, Col 15", and "INS".



The screenshot shows a Terminal window titled "vaibhav@vaibhav-VirtualBox: ~/LP5/Assign3" with the following output:

```
vaibhav@vaibhav-VirtualBox:~/LP5/Assign3$ export MPJ_HOME=/home/vaibhav/Downloads/mpj-v0_44
vaibhav@vaibhav-VirtualBox:~/LP5/Assign3$ export PATH=$MPJ_HOME/bin:$PATH
vaibhav@vaibhav-VirtualBox:~/LP5/Assign3$ javac -cp $MPJ_HOME/lib/mpj.jar arrSum.java
vaibhav@vaibhav-VirtualBox:~/LP5/Assign3$ $MPJ_HOME/bin/mpjrun.sh -np 4 arrSum
MPJ Express (0.44) is started in the multicore configuration
Enter 20 elements
Element 0 = 1
Element 1 = 2
Element 2 = 3
Element 3 = 4
Element 4 = 5
Element 5 = 6
Element 6 = 7
Element 7 = 8
Element 8 = 9
Element 9 = 10
Element 10 = 11
Element 11 = 12
Element 12 = 13
Element 13 = 14
Element 14 = 15
Element 15 = 16
Element 16 = 17
Element 17 = 18
Element 18 = 19
Element 19 = 20
Intermediate sum at process 3 is 90
Intermediate sum at process 2 is 65
Intermediate sum at process 1 is 40
Intermediate sum at process 0 is 15
Final sum : 210
vaibhav@vaibhav-VirtualBox:~/LP5/Assign3$
```