Logic Used:

Master Process

Master class initializes array by taking constraints from user and then randomly fills values from the range 0 – Array Size in the entire array, then takes input from user for value to search. All the data is then evenly distributed between all the worker processes and to make sure the distribution is even at the time of array size input, it is checked if size will be equally distributable, if not user is asked to enter it again.

The master process then uses an infinite while loop to wait for signals from worker processes. The loop only breaks if either value is found or each worker process sends signal that they could not find the values, for this a counter is used which counts the amount of worker processes which sent the signal of value not found in their set of values, if counter is equal to number of worker processes then all worker processes are asked to abort and loop breaks. Master sends 3 types of abort signals, 0 means continue searching, 1 means abort as some other process has found value and 2 means the worker process itself found value and can successfully terminate.

Worker Process

In the worker process another array is initialized of the distributed size that is allotted to it which is sent by master along with the values array should be populated it. Each worker process traverses its array and searches for the value sent by master process. Every time it checks a value it lets master know through '0' signal that 'I am still looking', if value is found signal sent is '1', and if value is not found and entire array is traversed signal sent is '2'.

The worker process only aborts if master tells it to, even if it has completed its traversal it will keep waiting for signal from master process, in this way all worker processes always relatively abort at the same time. In abort, 0 means keep going, 1 means some other process has found the value and 2 means you yourself found value and can now successfully terminate.

Use of MPI

In this assignment MPI was used to create master and worker processes and MPI_Send() and MPI_Recv() were used for communication between master and worker processes.

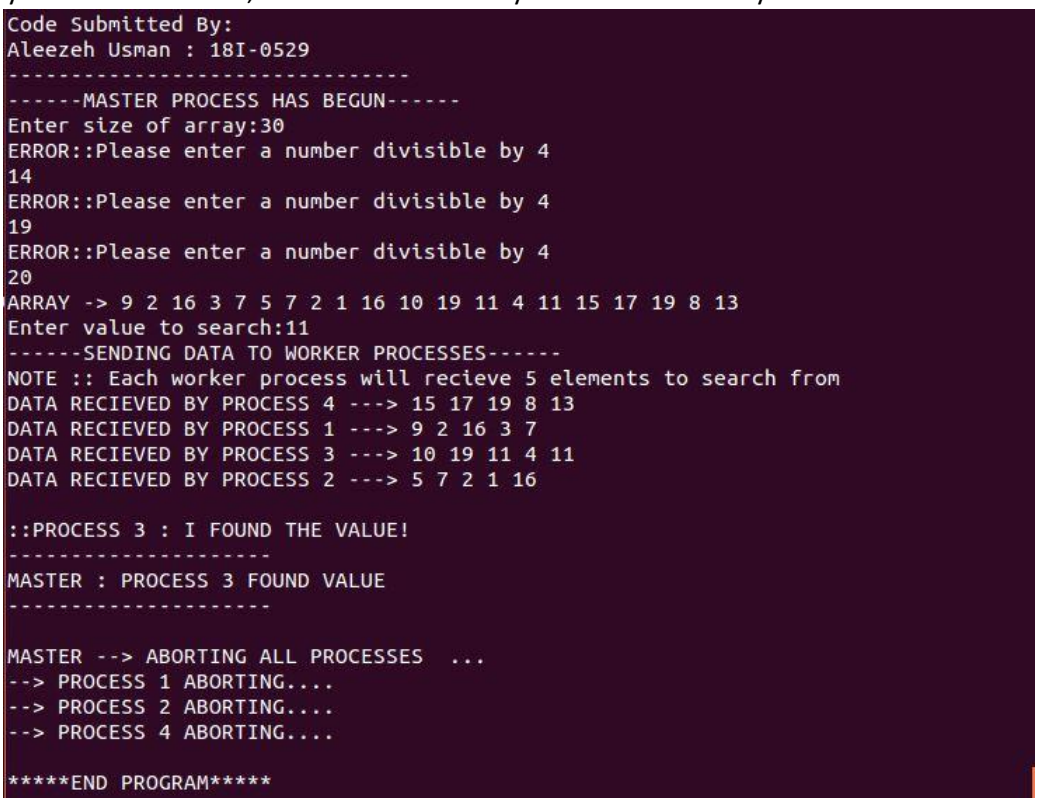
Use of OpenMP

OpenMP was also used for synchronization. #pragma omp critical{} and #pragma omp barrier{} were used to make sure critical section was properly defined so errors could be minimized and barriers were used to make sure processes began and ended at the same time and the entire program ran in sync.

Screen Shots

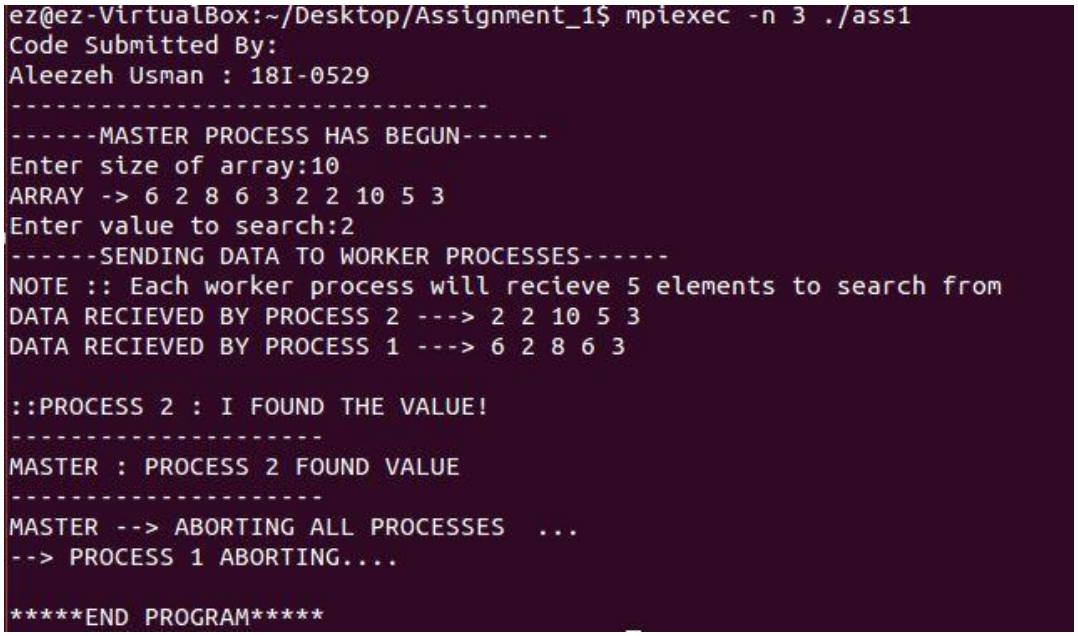
Screenshot 1: Input validation shown + Value found

Here number of total processes is 5 out of which 4 are worker processes so array elements need to be evenly divided between 4, therefore size of array must be divisible by 4.



Screenshot 2: Value found using 2 worker process.

Here given value was present in both processes but because it is found by process 2 first it returns it to Master and other process aborts its search



Screenshot 3: Value not found using 5 worker processes

```
Code Submitted By:
Aleezeh Usman : 18I-0529
--------------------------------
------MASTER PROCESS HAS BEGUN------
Enter size of array:25
ARRAY -> 9 10 5 17 17 24 20 25 14 16 1 9 22 12 13 18 8 6 21 14 1 12 8 22 5
Enter value to search:32
------SENDING DATA TO WORKER PROCESSES------
NOTE :: Each worker process will recieve 5 elements to search from
DATA RECIEVED BY PROCESS 4 ---> 18 8 6 21 14
DATA RECIEVED BY PROCESS 2 ---> 24 20 25 14 16
DATA RECIEVED BY PROCESS 5 ---> 1 12 8 22 5
DATA RECIEVED BY PROCESS 1 ---> 9 10 5 17 17
DATA RECIEVED BY PROCESS 3 ---> 1 9 22 12 13
<<<< ERROR :: VALUE WAS NOT FOUND >>>>

MASTER --> ABORTING ALL PROCESSES  ...
--> PROCESS 2 ABORTING....
--> PROCESS 1 ABORTING....
--> PROCESS 3 ABORTING....
--> PROCESS 5 ABORTING....
--> PROCESS 4 ABORTING....

*****END PROGRAM*****
```

Screenshot 4: Value found using 2 worker processes

Here both processes have value at exact same spot but value is only sent back by one of the process and the other is aborted, which process sends value back is entirely based on which process is scheduled first by CPU

```
Code Submitted By:
Aleezeh Usman : 18I-0529
--------------------------------
------MASTER PROCESS HAS BEGUN------
Enter size of array:10
ARRAY -> 10 1 7 4 3 9 1 6 7 6
Enter value to search:1
------SENDING DATA TO WORKER PROCESSES------
NOTE :: Each worker process will recieve 5 elements to search from
DATA RECIEVED BY PROCESS 1 ---> 10 1 7 4 3
DATA RECIEVED BY PROCESS 2 ---> 9 1 6 7 6

::PROCESS 1 : I FOUND THE VALUE!
---------------------
MASTER : PROCESS 1 FOUND VALUE
---------------------

MASTER --> ABORTING ALL PROCESSES  ...
--> PROCESS 2 ABORTING....

*****END PROGRAM*****
ez@ez-VirtualBox:~/Desktop/Assignment_1$
```