1) For MPI programs, MPI_Send must use receiving process id and MPI_receive should use sender's id. In this program they are corrected.
Also data types must be same at both sender's and receiver's end, so corrected those as well.

2) Every MPI Program must use MPI_Init(&argc, &argv) and MPI_Finalize() to initialize MPI library and to clean up all MPI state. These two calls are missing in this program.

3) MPI_Send and MPI_Receive must have matching tags for successful communication. "tag" represents user defined type for the message. This program is corrected by putting tag as 0 for all the calls.

4)Given program in problem can cause deadlock because thread in first section 1 aquiring lock1 followed by big loop and then trying to aquire lock2. Meanwhile thread in section 2 is acquiring lock2 followed by big loop and then trying to acquire lock1. This is the one of sceanrios where deadlocks can happen, one thread is having lock and trying to acquire another lock which is already loced by another thread which is waiting for first lock acquired by former thread. These deadlocks can be prevented when we acquire locks sequentially. By making Thread 1 to first get lock1 and then try to acquire lock2 which is same route for Thread 2 to acquire locks, we can prevent deadlock in this situation.

5) This program is printing different sums on every execution because 'sum' is shared variable and multiple threads are accessing it to modify its value. By using 'reduction' we can get consistent and correct answer all the time.