Pseudocode for parallel_gauss.cpp

by Karlo Martinez Martos
and Cassandra Schaening-Burgos

Let MATRIX be represented as an array of size n by m;

```
eliminate(pivot  row, row, m, pivot_position)
        inverse = additive inverse of row[pivot_position]
        parallel for i = 0 → m
                row[i] = row[i] + pivot_row[i] * m
                row[i] = row[i] mod p

For i = 0 → n:
        processor 0 broadcasts the pivot row MATRIX[i]
        for j = i + 1 → n:
                a BARRIER in each iteration synchronizes the processes
                processor 0
                        sends row MATRIX[j] to processor  j mod numprocs
                        receives reduced row MATRIX[j]
                each processor !=0
                        receives a row MATRIX[j]
                        eliminate( pivot row, row, m, i)
                        send row back to processor 0
For i = n -1 → 1
        processor 0 broadcasts pivot row MATRIX[i]
        for j = i – 1 → 0
                processor 0
                        sends row MATRIX[j] to another processor
                        receives reduced row MATRIX[j]
                each processor != 0
                        receives a row MATRIX[j]
                        eliminate(pivot row, row, m, i)
                        send row back to processor 0
```