Katelyn Jaing
Melissa Riddle
Hector Medina
CPSC 479-01
Project 1: Pseudocode for concurrent two leader election algorithm

---

To run (N = greater than 5, less than 20) :

```
mpicc project1.c
mpirun -n N ./a.out
```

```
#include <stdio.h>, <mpi.h>, <stdlib.h>, <time.h>
int main( int argc, char *argv[] ){
        temp, rank, size, array[ 2 ];
        MPI_Init( &argc, &argv );
        MPI_Comm_rank( mpi_comm_world, &rank );
        MPI_Comm_size( mpi_comm_world, & size );
        if size is less than 5 or size is greater than 20 and rank is 0:
                printf( “Error” );
                MPI_Abort( MPI_Comm, error_code );
        else if rank is not 0:
                MPI_Recv( &array, 2, int, rank-1, 0, mpi_comm_world, mpi_status_ignore );
                printf( “Process %d received even %d odd %d from process %d” );
                temp = rand();
                if temp is greater than 100: temp = temp % 100;
                else if temp is less than 10: temp = temp + 10;
                temp = temp + 1000 + ( rank * 100 ); //concatenate
                if temp is even: if temp is greater than array[ 0 ]: array[ 0 ] = temp;
                else: if temp is greater than array[ 1 ]: array[ 1 ] = temp;
                MPI_Send( &array, 2, int, rank+1, 0, mpi_comm_world);
                if rank != size-1: printf( Process %d sent even %d odd %d to process %d” );
                else: printf( “Process %d sent even %d odd %d to process 0” );
        else:
                temp = rand();
                if temp is greater than 100: temp = temp % 100;
                else if temp is less than 10: temp = temp + 10;
                temp = temp + 1000 + ( rank * 100 ); //concatenate
                if temp is even: array[ 0 ] = temp, array[ 1 ] = 1;
                else: array[ 0 ] = 0, array[ 1 ] = temp;
                MPI_Send( &array, 2, int, rank+1, 0, mpi_comm_world);
                printf( “Process 0 sent even %d odd %d to process %d” );
        if rank is 0:
                MPI_Recv( &array, 2, int, size-1, 0, mpi_comm_world, mpi_status_ignore );
                printf( “Process 0 received even %d odd %d from process %d” );
                printf( “President: %d, Vice President: %d” );
        MPI_Finalize();
        return 0;
}
```