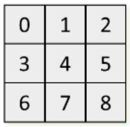
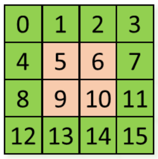
Assignment 1 (100 marks)

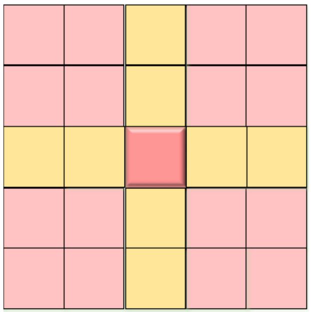
Write a program to perform halo exchange (refer to Lectures 12 and 13) with neighbouring processes. The boundary processes need not exchange data in case of non-existing neighbours (for e.g. consider the process grid on the right, process 0 does not have a top and left neighbour, process 1 has three neighbours, and so on).

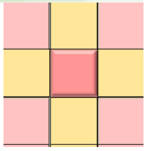


Assume Px X Py decomposition of processes. Every process owns a subdomain of data (doubles). For example, a sub-domain is shown in the right figure, every process has 16 data points and 4 halo regions (shown in green). The halo regions need to be exchanged with four (or less as the case may be) neighbouring processes (top, bottom, right, left) at every time step.



Every process performs communication for the boundary points/cells followed by a stencil computation. Next time step value at a point/cell P is computed as the average of the four neighbouring points/cells of P (left, right, top, bottom) in case of 5-point stencil. E.g. value (P, t+1) = [value (P_left, t) + value (P_right, t) + value (P_top, t) + value (P_bottom, t) + value (P, t)] / 5. Similarly for 9-point stencil, the average of the eight neighbouring points are used.





The assignment is to compare the performance of data exchange for the given domain sizes and process counts for (i) 5-point stencil and (ii) 9-point stencil using MPI_Pack/MPI_Unpack and MPI_Send/MPI_Recv for communication.

Perform the experiments for the following configurations.

```
for execution in 1 to 3 // repeat each configuration 3 times
 Run on PRUTOR (a script will automatically run the below configurations)

 for P (number of processes) in 12 [Use Px=4 and Py=3]
   for N (double data points per process) in 512^2, 2048^2
     for stencil in 5, 9
    mpirun –np P –f hostfile ./halo Px N <num_time_steps> <seed> <stencil>
```

Execute with num_time_steps = 10 (this is the third argument), i.e. computation and communication is done for 10 time steps.

Every process must randomly initialize the N data points that it owns at time step 0 using the 'seed' value given in the fourth argument and the following equation.

```
srand(seed*(myrank+10));
data[i][j] = abs(rand()+(i*rand()+j*myrank))/100;
```

Use 6 processes per node. Time the entire data exchange for all the time steps per data size per stencil configuration. Output the time only (as returned by the MPI_Wtime() function).
<time1>
For the above run, there must be four output lines (times).

Plot the time (in seconds) for each data size per stencil configuration. Use boxplots (from the 3 executions) for every data point in the plot. Time in seconds in y-axis and (N, stencil) in x-axis. You should report the total time taken by the main halo exchange function, excluding initialization.

Execution and submission instructions

1. Submit your code on PRUTOR. Do not use the arguments pragma while collecting timing information and your final submission.
2. Submit GroupXY.pdf to hello.iitk.ac.in (Assignment 1). It should necessarily contain the source code ('src.c'), 'readme.pdf', plot script and data file. The source code should be well documented. The 'readme.pdf' should contain a simple explanation of your code, the timing plot, your detailed observations regarding performance from the plot, any optimizations you have used within the code (maximum 4 pages).

Clearly mention your group number, member details (names, roll numbers, IITK email IDs) in your readme.pdf.

The above submissions must be made by only one group member. We will consider the last submitted code on PRUTOR (before the deadline) as the final code.

Due date: 24-03-2024 (There will be NO extensions)