Homework 7, due: 03/31

MATH 9830, Spring 2015 Timo Heister, heister@clemson.edu

0. Before you start:
   - PLEASE make sure you send me the code via email (you do not need to print any code).
   - Print (or email a pdf) with your output (pictures, text), explanations, and drawings.
1. Hanging nodes and constraints:
   - Familiarize yourself with adaptive refinement and hanging nodes in `deal.II`. Relevant program: step-6, relevant videos: 15,16,17.
   - In step-6, switch to vtk output and create a plot of the solution once with the `constraints.distribute()` call in the code and once without (just comment it out). Describe in your own words what this call does.
   - Create a new (short) program that creates a simple mesh with 7 cells and two hanging nodes (refine a 2d hypercube once and then refine the first cell one more time). This mesh is also used in the tutorial videos. Create the `ConstraintMatrix` for a Q1 and a Q2 finite element and print it to the screen (it should be 2 constraints for Q1).
   - Output the positions of the support points of the degrees of freedom with the help of the function `DoFTools::map_dofs_to_support_points`. With this information, create a (hand-drawn) mesh with the dof indices marked on there (one picture for Q1, one for Q2).
   - Can you explain why there are constraints in the form $x_i = x_j$ for Q2?
   - List several examples of things you can do with contraints.
2. MPI Collectives ("let's play dice"):
   - In a loop, let each process roll a dice (draw a random number between 1 and 6 using `1+rand()%5` (initialize the random number generator at the start with `srand(time(NULL)+rank);`). Exit the loop on each process if a) nobody rolls a 1 or a 2 (hint: minimum of rolls is?), and b) the average value is exactly 4 (hint: or the sum is?).
   - `Gather` the rolls from the last successful round on rank 0 and print them.
   - Only use MPI collectives and only print things on rank 0. Your final output should be something like:

```
$ mpirun -n 6 ./main
Running with 6 ranks. Trying to roll a sum of 24 ...
It took 213 rolls to get: 6 3 6 3 3 3
```

   - Good luck!