Exercise 2

*Tasks marked with a * are assessed coursework. Hand in your solutions to these via email to* `rn@ic.ac.uk`. *(Resit students do not need to submit coursework.) Use the subject line "*`C++ CW: surname_firstname_CW2`*", where* `surname_firstname_CW2.cpp` *is the attached file that contains your solution. The course will be assessed based on 5 pieces of coursework (25%) and an end of term driving test (75%). Your submission must be* ***your own work*** *(submissions will be checked for plagiarism), and it should compile (and run) with the GNU C++ compiler* `g++`. *The deadline for submitting the coursework is 10pm on* **10/02/2019**.

Continue to use the STL container `vector<>`. Although this is not(!) necessary, and in no way the only possible solution strategy, you could use two-dimensional `vector<vector<> >` in your code for the following exercises. A quick and easy way to do so is shown here.

```
int nrows = 4, ncolumns = 3;
vector<vector<double> > mat;                    // note the space in '> >' !
vector<double> row(ncolumns);                   // row = (0,0,0)

mat.insert(mat.begin(), nrows, row);            // mat = (row,row,row,row)
mat[1][2] = 2;
for (int i=0; i<mat.size(); ++i)  {
  for (int j=0; j<mat[i].size(); ++j)  cout << mat[i][j] << " ";
  cout << endl;
}
```

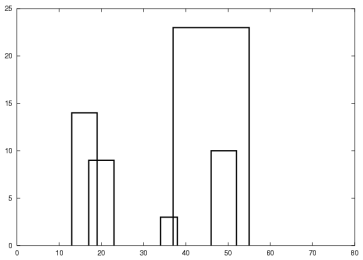
1. *Oozing oil*

   Assume the following simplified model for oil oozing through soil. The ground is described by a rectangular box of dimension 80×60 with integer coordinates. The y-axis is oriented downwards, the x-axis is oriented to the right as usual. Initially the oil is distributed in the upper most layer only. More precisely, initially oil can only be found at all the coordinates (x, y), where $y = 1$ and where x is an odd number such that $20 \leq x \leq 60$. The oozing of the oil is modelled as follows. If there is oil at position (x, y) then from here it reaches the next lower layer at position $(x-1, y+1)$ with probability p $(0 < p < 1)$, and independently from that also at position $(x+1, y+1)$, also with probability p. Note that the amount of oil may change from one layer to the next. We say that the oil oozed through if at the end one can find oil in the lowest layer $(y = 60)$.

   Write a program that reads in the probability p, then graphically simulates the oozing of the oil on screen from the first layer $(y = 1)$ to the lowest layer $(y = 60)$ and states whether some oil managed to ooze through to the lowest layer. What is the smallest values of p that regularly lets oil ooze through to the lowest layer?

2*. *Skyline*

   Assume that you are given the coordinates of all the major buildings of a city on a 80x25 grid, i.e. $0 \leq x \leq 80$ and $0 \leq y \leq 25$. Here we assume for simplicity that all the buildings are cuboids, hence it suffices to know the upper left and upper right corner of each building. The example on the right shows the buildings for the following sequence of coordinates: (17,9), (23,9), (13,14), (19,14), (46,10), (52,10), (37,23), (55,23), (34,3), (38,3).



   Write a program that reads in a list of building coordinates from the file `skyline.in` and returns the ***skyline*** of the city as a *sorted sequence of distinct vertices* in the file `skyline.out`. In the above example the skyline for the first two buildings is given by the unique (!) sequence (13,0), (13,14), (19,14), (19,9), (23,9), (23,0). Your program should not ask for any user input and the file `skyline.out` must only contain numbers, presented in two columns: the x- and y-coordinates of the sorted list of vertices. Hence the following `gnuplot` and MATLAB commands can be used to plot the skyline:

```
gnuplot> plot "skyline.out" w lp
MATLAB> s = load('skyline.out'); plot(s(:,1), s(:,2), 'b.-');
```

   Test your program with the sequence of building coordinates that can be downloaded from `www.ma.ic.ac.uk/~rn/teaching/skyline.in`.

   [*Hint: Natural class definitions for this task would be e.g.* `Point`, `Corner` *or* `Building`.]