1. Write a C++ program to simulate one-dimensional diffusion. The set of discrete points extends $-L < x < L$, so if there are M points, the lattice step is $h = 2L/M$. The constants are $L = 50$, $M = 100$, diffusion constant $D = 1$. The time step $\tau = ch^2/(2D)$, where c is read from the console.
   First, write function `init()`, that puts a Gaussian distribution of particles around $x_0 = 0$, with width $\sigma = L/30$. The density in point i is initially
   $$\rho_i = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-x_0)^2}{2\sigma^2}} ; \tag{1}$$
   Next, write a function `evolve()` that updates the density. Turning on diffusion, the density evolves to the next time step as
   $$\rho_i(t+\tau) = \rho_i(t) + \frac{D\tau}{h^2}(\rho_{i+1}(t) - 2\rho_i(t) + \rho_{i-1}(t)) \ , \tag{2}$$
   where the second derivative was discretized according to
   $$\rho_i''(t) \approx \frac{1}{h^2}(\rho_{i+1}(t) - 2\rho_i(t) + \rho_{i-1}(t)) \ . \tag{3}$$
   Using $c << 1$, store the density at suitable times to file `density.dat` in format "time point density". If you like, plot it using, say, `gnuplot` (start gnuplot, type `sp 'density.dat' w pm3d`) . Don't write density to file at all times, it's too much.
   I suggest you have the density profile stored as `vector<double> rho`. If you wish, try any initial density profile you like are welcome. Be ware, that eventually particles diffuse out of your simulation box.