Let c and z_0 be complex numbers. We define the following recursion:

$$z_n = z_{n-1}^2 + c$$

This is a dynamical system known as a quadratic map. Given different choices for parameter c and the initial value z_0 the recursion leads to a sequence of complex numbers $z_1, z_2, \ldots$ known as the orbit of z_0. This dynamical system is highly *chaotic*, meaning that depending on the selected c and z_0, a huge number of different orbit patterns are possible.

Suppose that we fix the parameter c. In such cases, most choices of z_0 tend towards infinity (i.e. $|z_n| \to \infty$ as $n \to \infty$). For some z_0 (this depends a little on c as well), however, the orbit is stable, meaning that it goes into periodic loop; and finally there are some orbits, that seem to do neither, dancing around the complex space apparently at random.

In this assignment, your task is to you write a MATLAB script that visualizes a slightly different set, called the filled-in Julia set (or Prisoner Set), denoted K_c , which is the set of all z_0 with orbits which do not tend towards infinity. The "normal" Julia set would be the edges of of K_c.

a) It is known that if the modulus of z_n (i.e. $|z_n|$) becomes larger than 2 for any n, the sequence will tend to infinity. The value of n for which this becomes true is called the 'escape velocity' of a particular z_0. Write a function that returns the escape velocity of given z_0 and c. Note you cannot test the recursion for all n: but rather you should select an upper bound N, so that if $|z_n| < 2 \, \forall n < N$, the function should return N. This allows you to avoid infinite loops.

b) Then write a function that takes $c, z_{\max}$ and N as arguments. The function will define a square in complex plane of complex numbers with real part between $-z_{max}$ and z_{max}, and imaginary part between $-z_{max}$ and z_{max}, and discretise it into a 500×500 grid. It will then compute the escape velocity of every element in the grid using the function you wrote previously, and the parameters c and N. Save the escape velocities to a matrix M; remember to preallocate.

c) Visualize your fractal using `imagesc(M)`. You may also want to try `imagesc(atan(0.1*M))`.