1. Given a point cloud in 2D, so basically a list of (x and y) coordinate pairs and **a** length scale
2. Any points that are within this length, should be considered as belonging to the same cluster. (Please note that if point A and B are within this length scale, and so are points B and C, even though points A and C may **not** be within the length scale, they **still** belong to the same cluster. Here points A, B and C are just for explanation.
3. How many clusters exist?

In the image below, I provide an example that contains just 5 points, and 5 distinct length scales. Note, that I am providing 5 length scales just for the sake of explanation, not for you to consider multiple length scales. You need to consider just a single, arbitrary length scale. I used different colors for each length scale. Note how the red is very small, and the purple is the largest. The number next to the length scale is the number of clusters that this length scale results for the 5 points in the image.

In your solution, you should consider the following:

1. The point cloud may contain millions of points
2. Distance calculation between points is known to be expensive. You should not try to care about this bottleneck.
3. Basically, you should develop a segregation algorithm.
4. Please develop a solution in C++.

