**DEPARTMENT OF INFORMATION TECHNOLOGY, NITK SURATHKAL**

**IT 301 PARALLEL COMPUTING**

**PC LAB 7**

**Date: 30 September 2020**

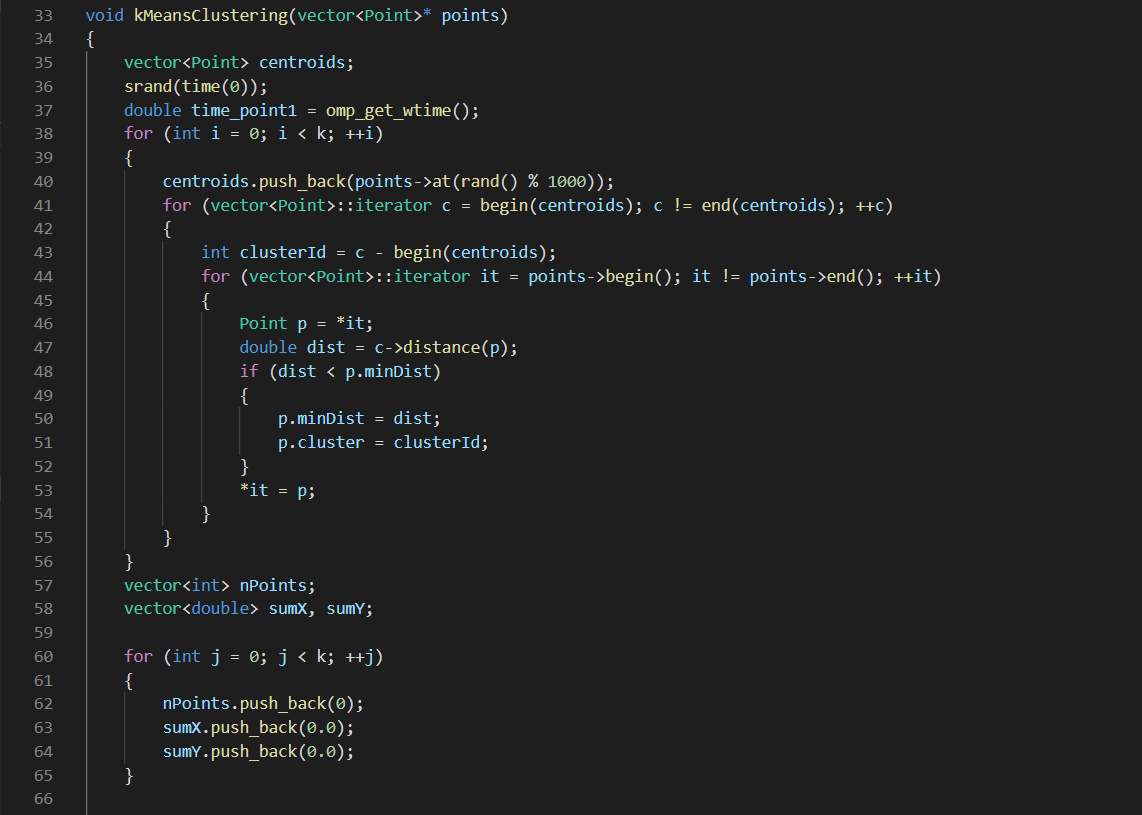
**Submission Date : 7th October 2020**

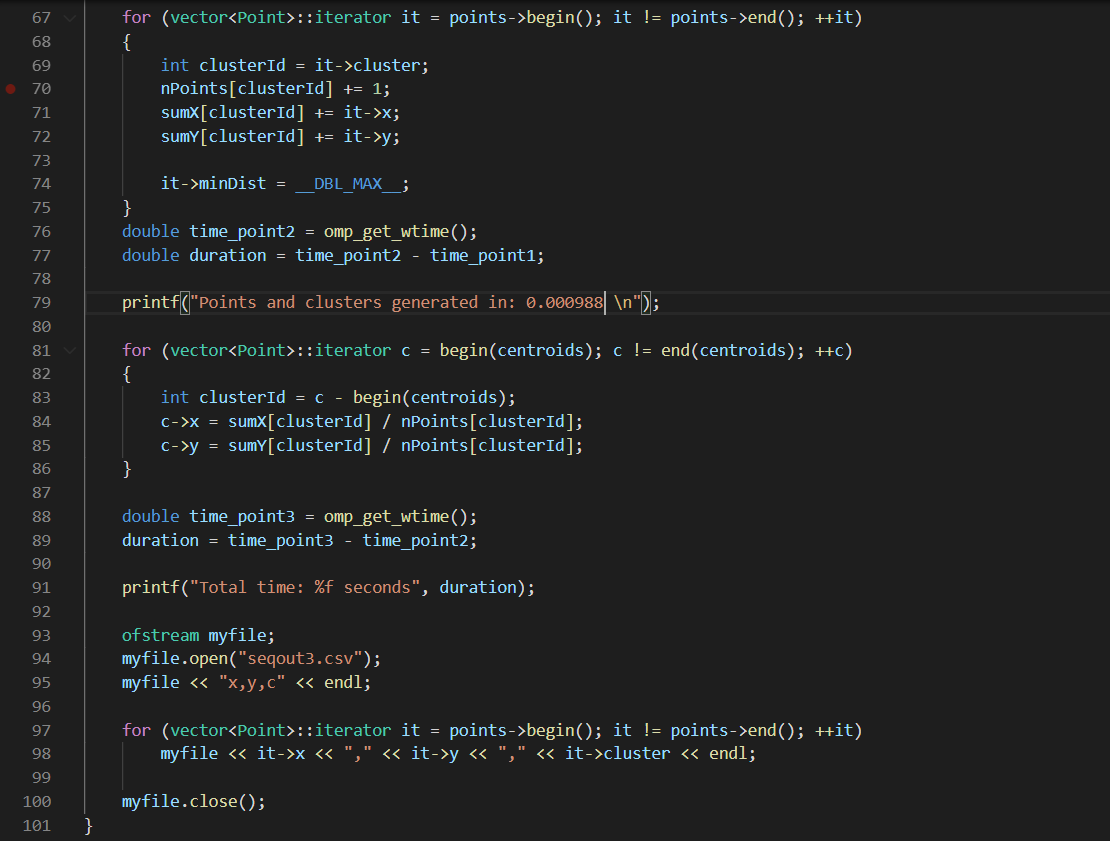
**Name**: Chinmayi C. Ramakrishna

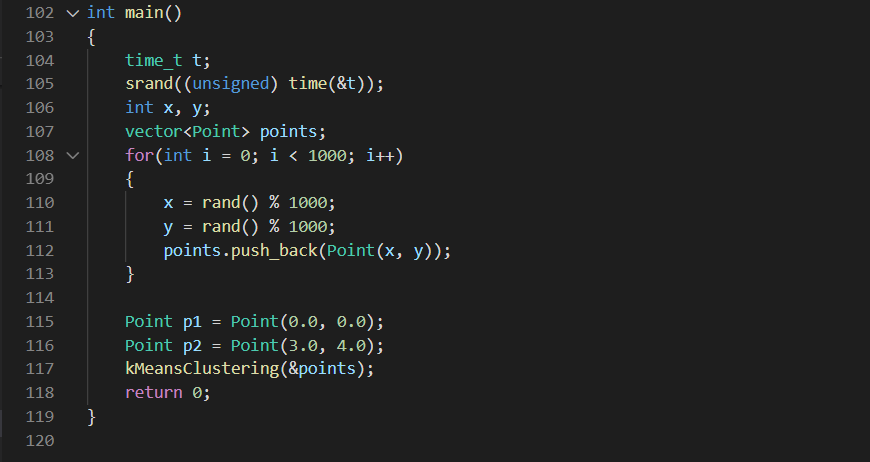
**Roll No.:** 181IT113

**Sequential k-means Clustering**

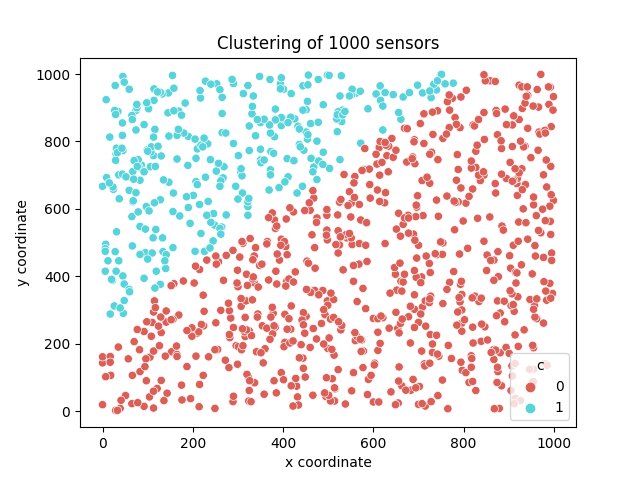




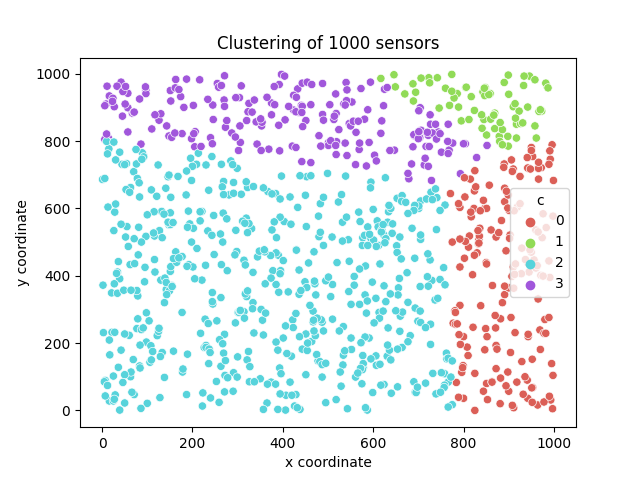




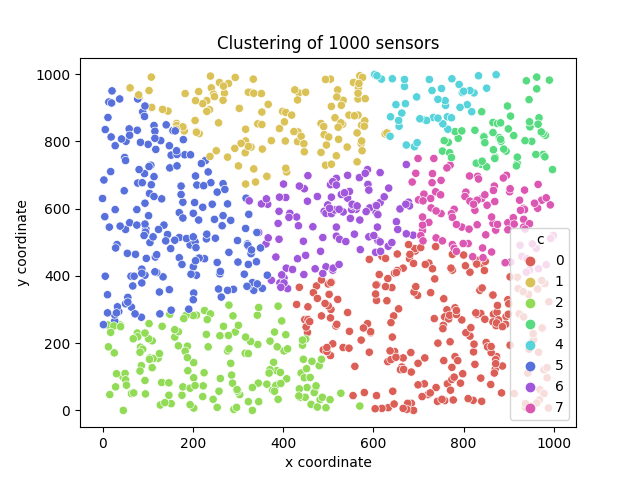
**Graphs:**



K = 2

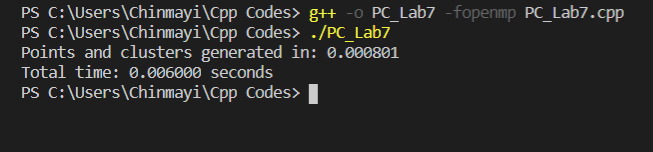


K = 4

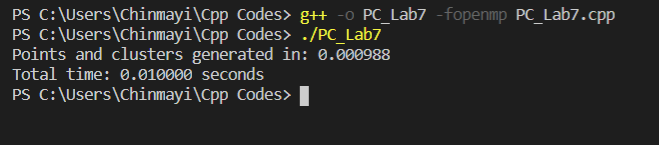


K = 8

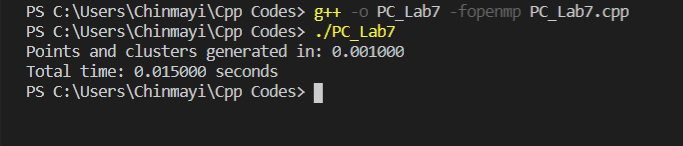
**Output:**



K =2

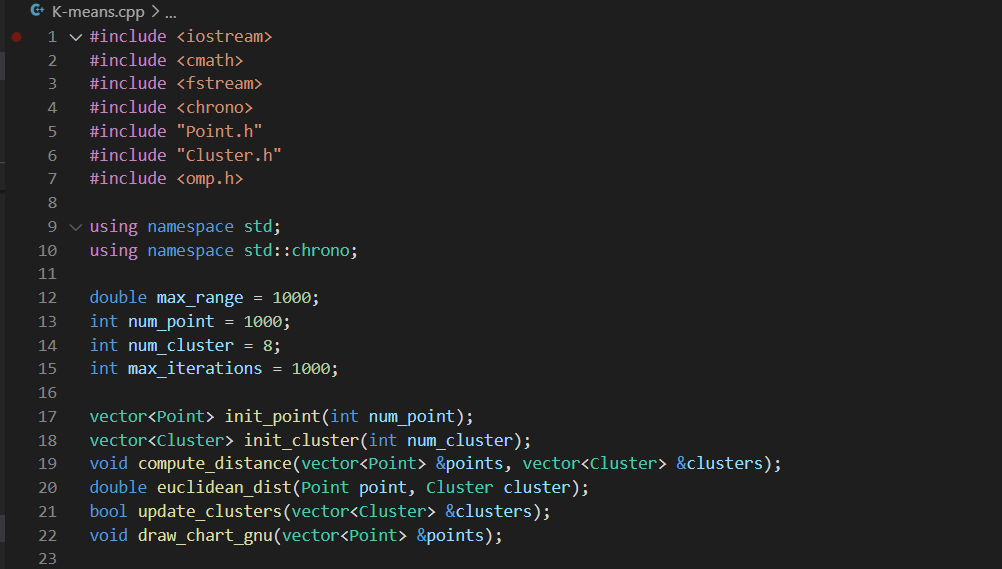


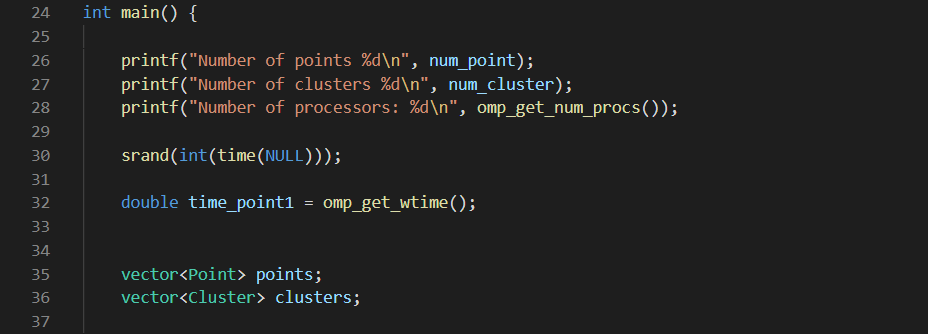
K = 4

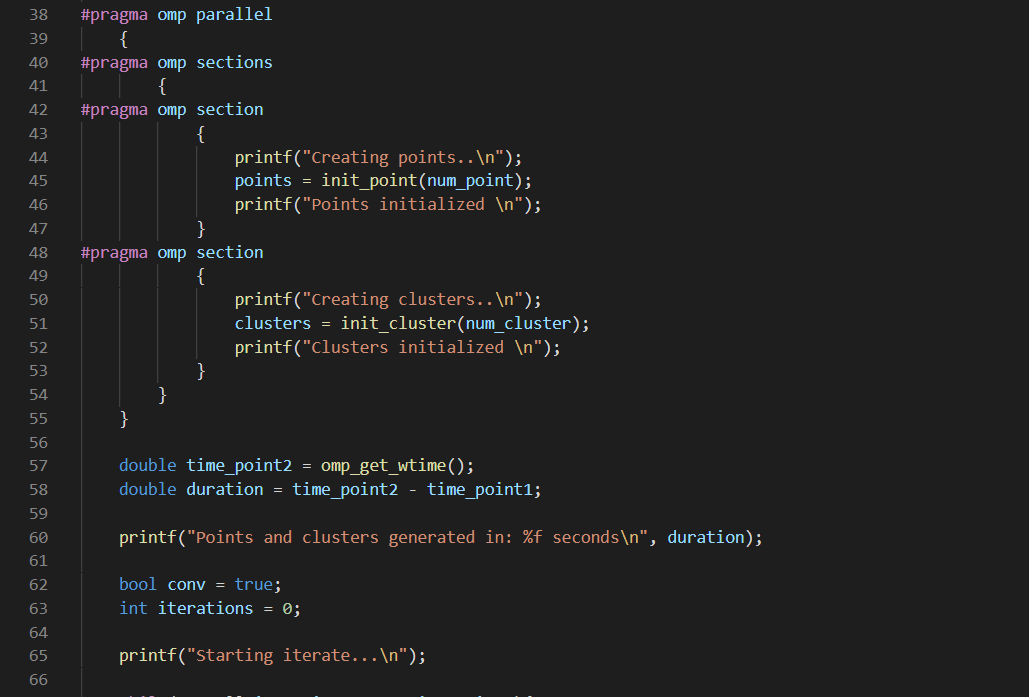


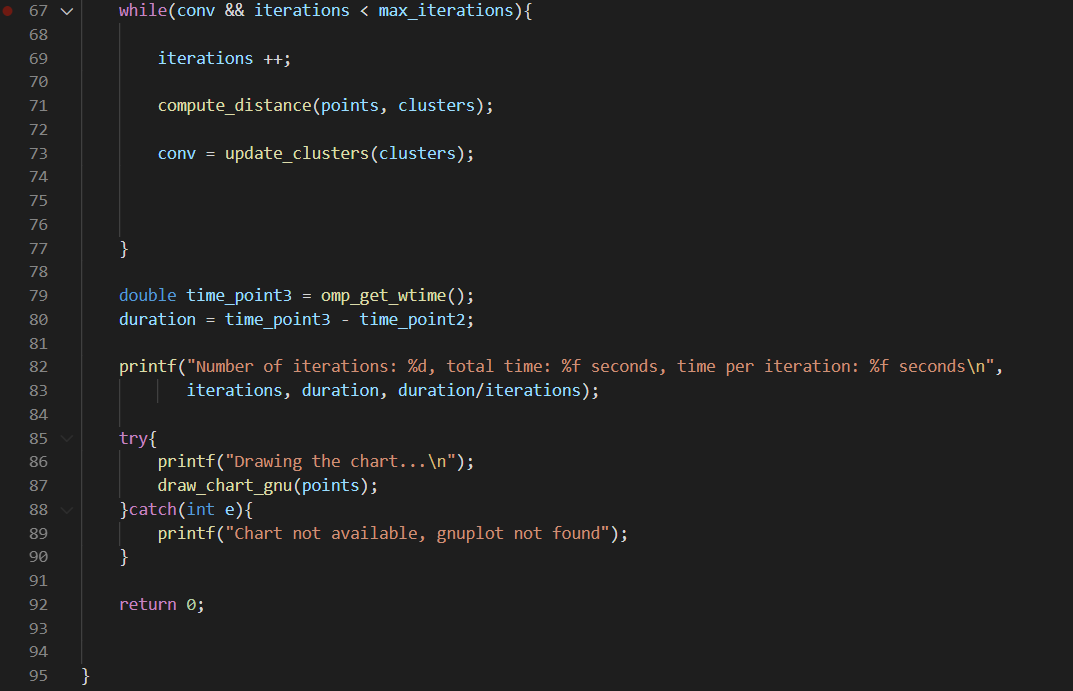
K = 8

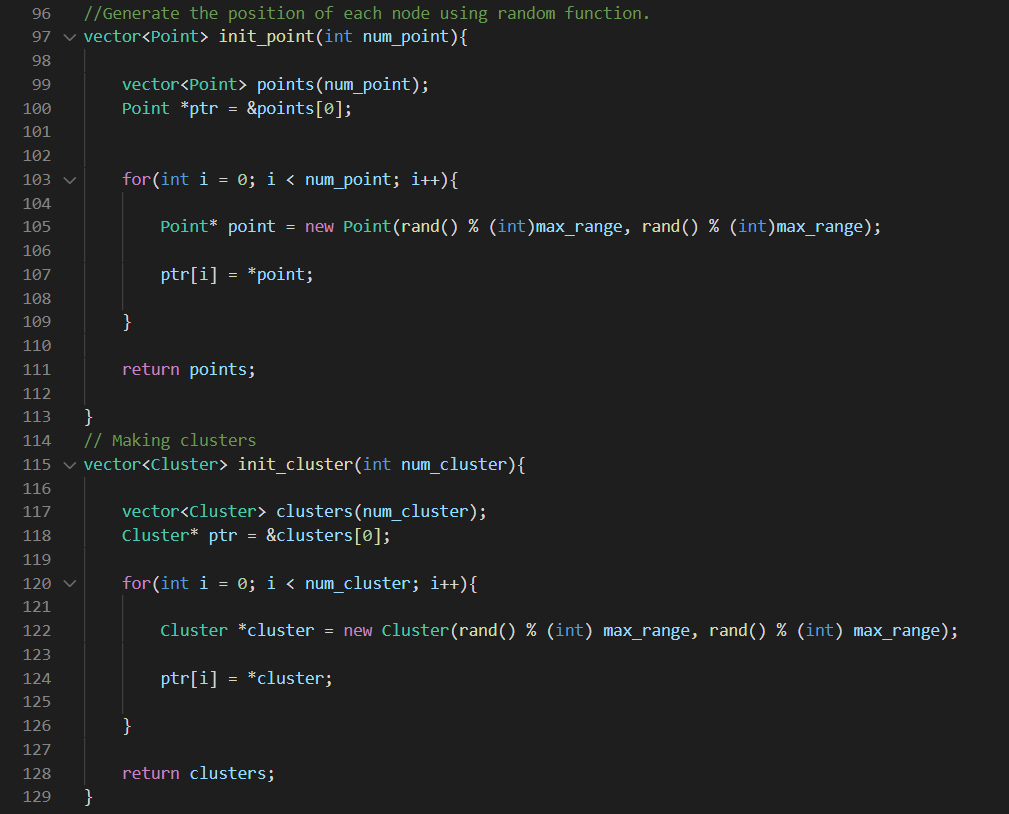
**Parallel K-means Clustering:**

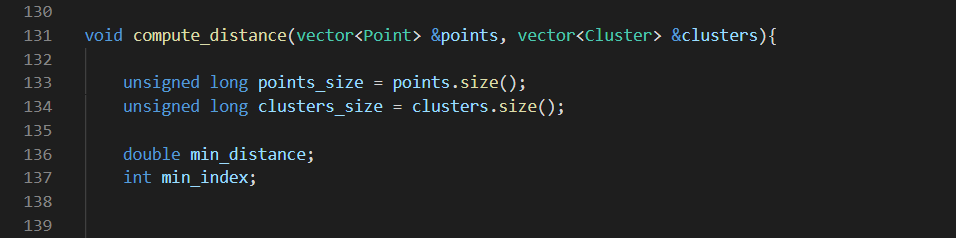


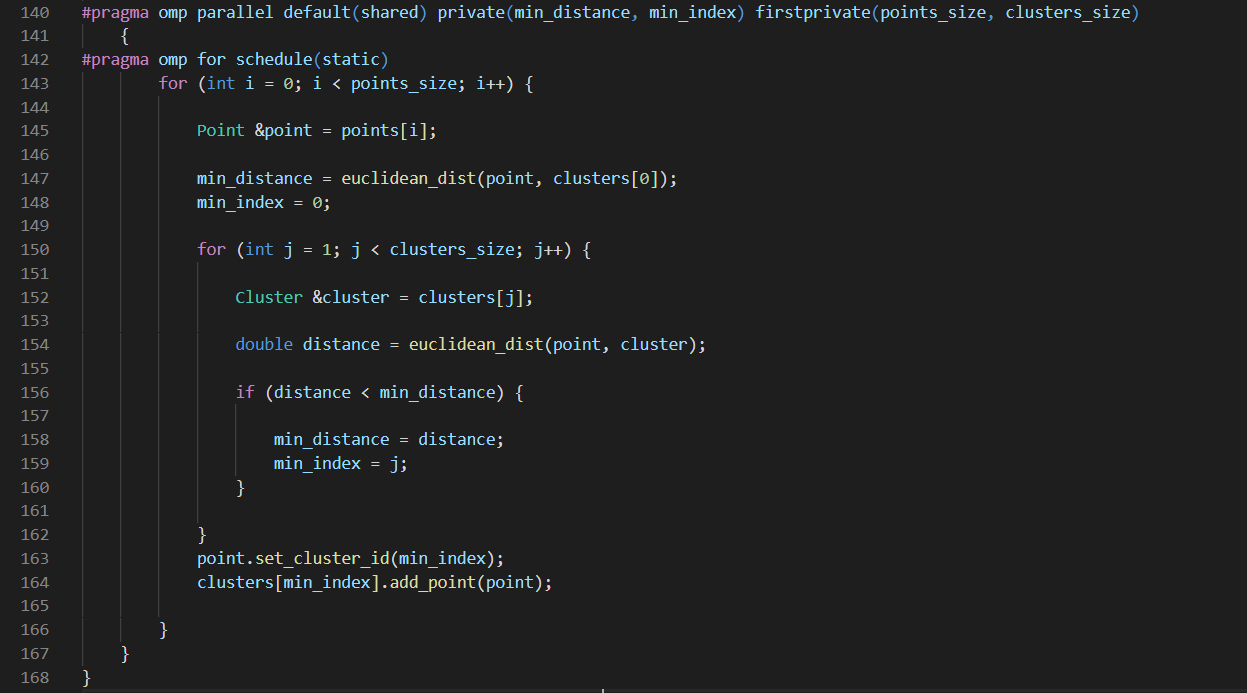


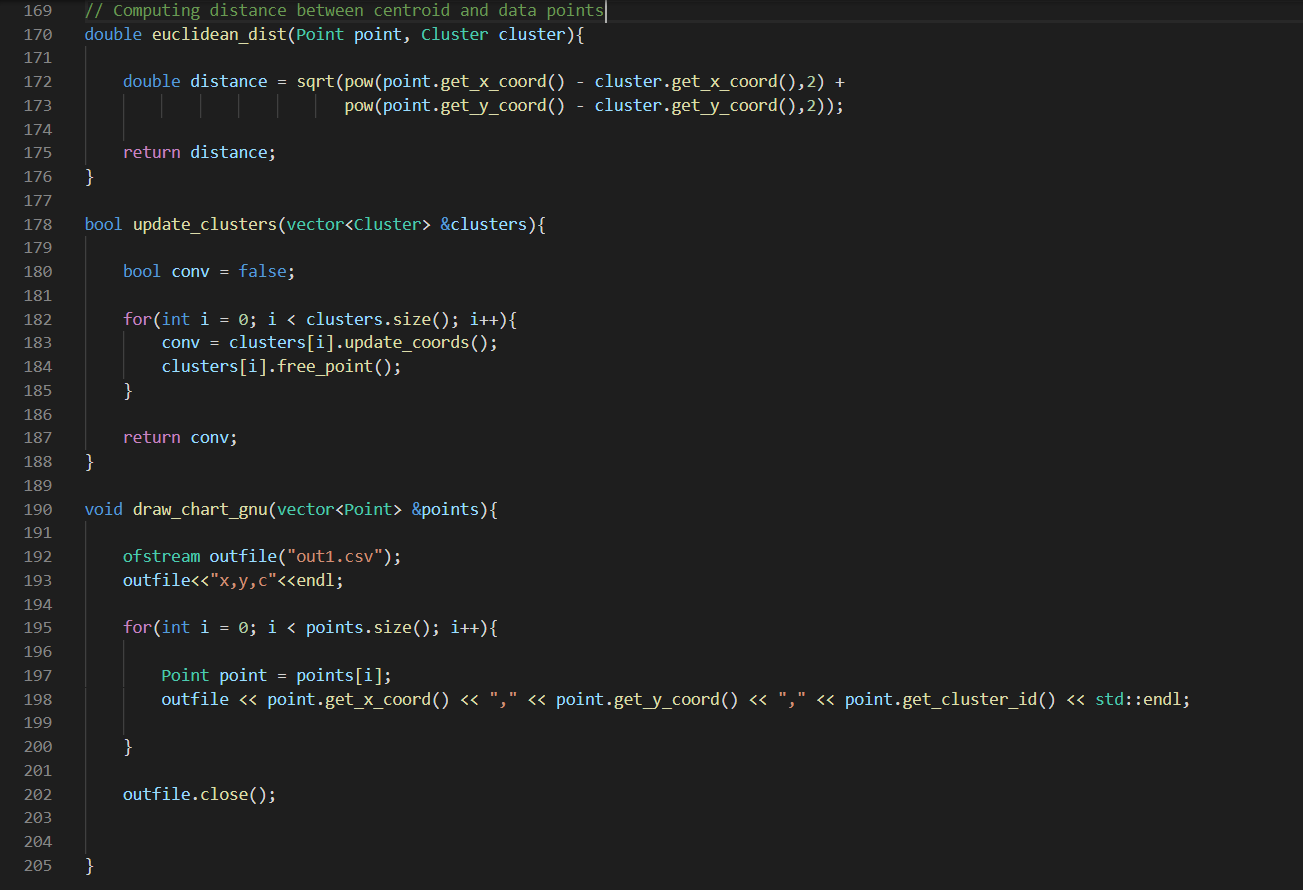






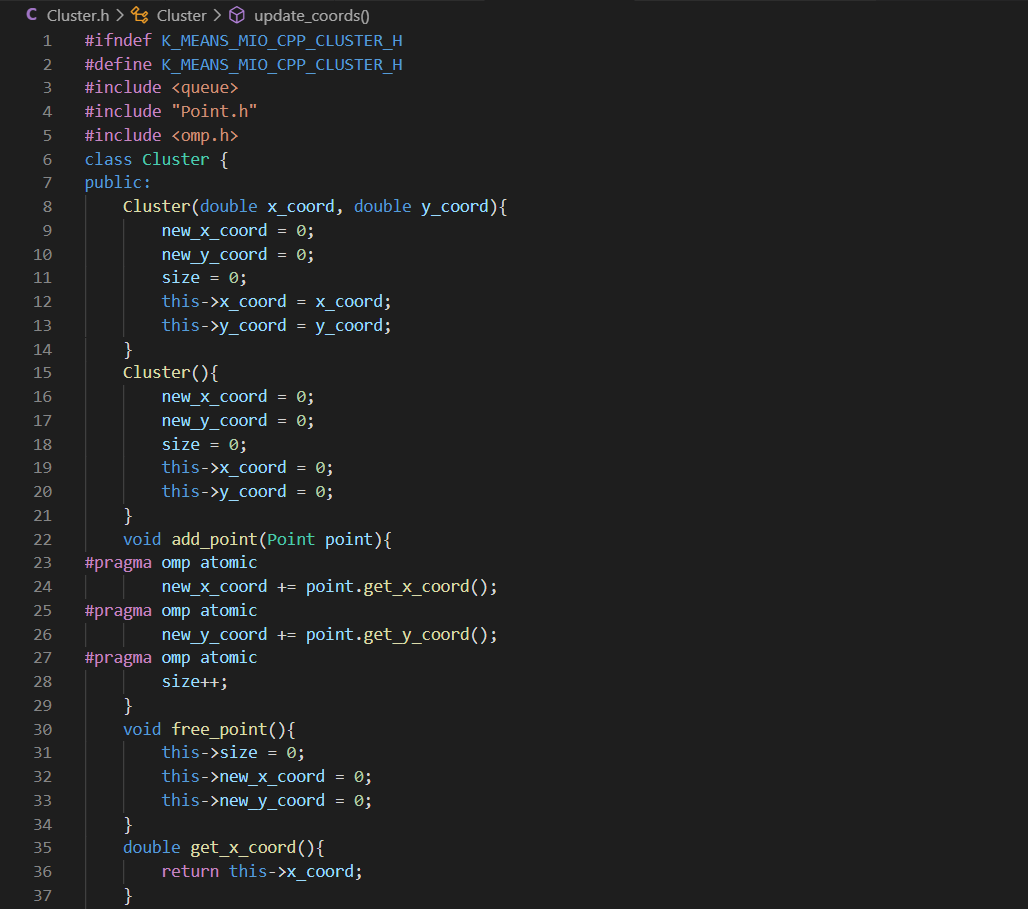


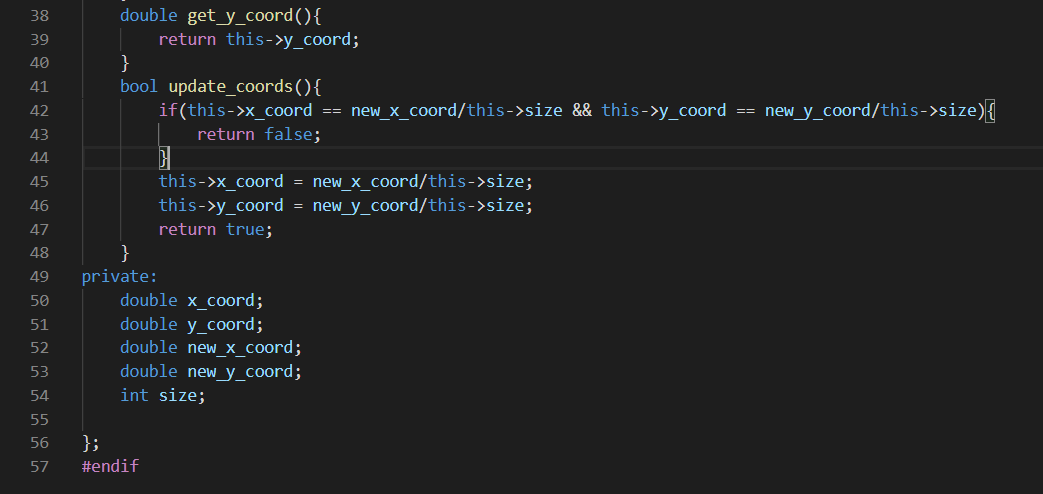






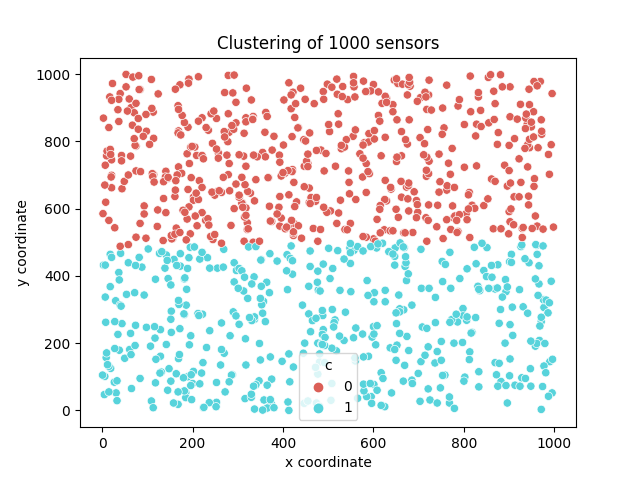
**Point.h**



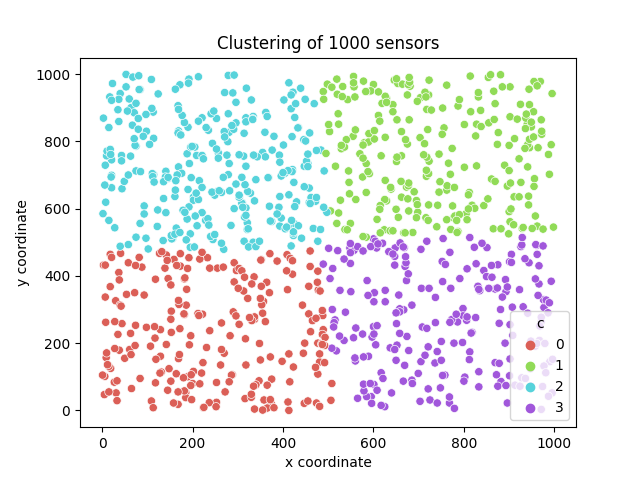


**Cluster.h**

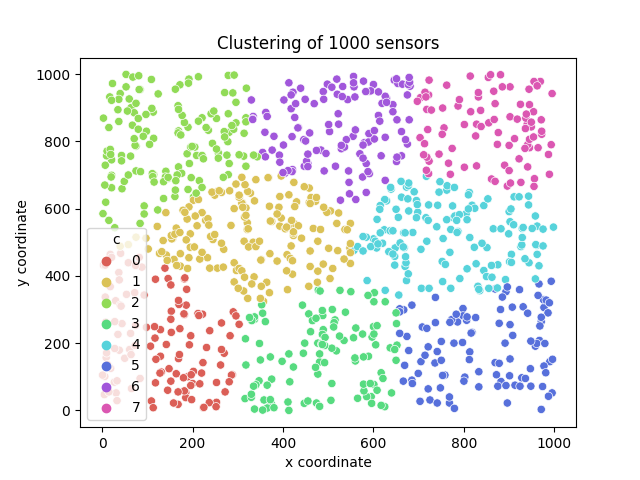
**Graphs:**

****

K = 2

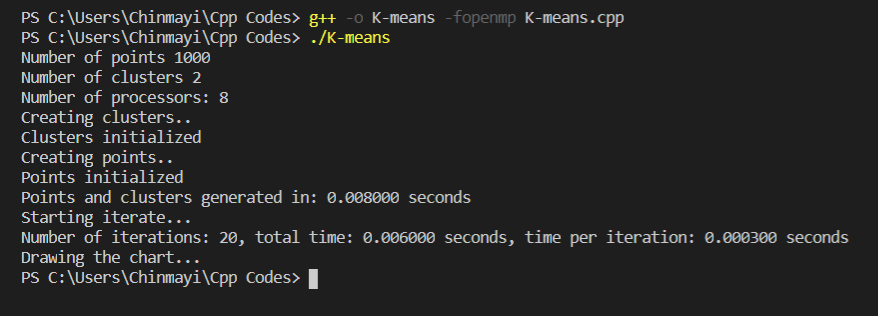
****

K = 4

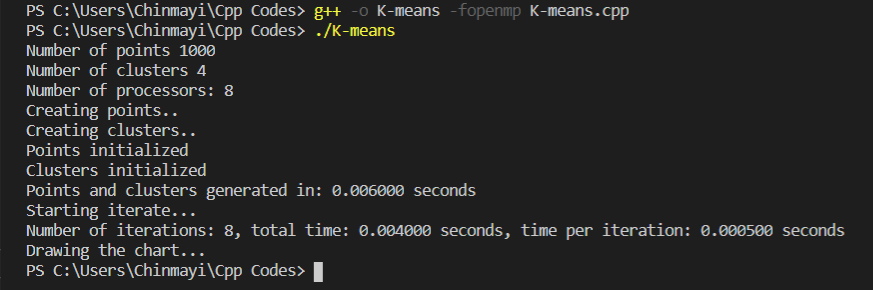
****

K = 8

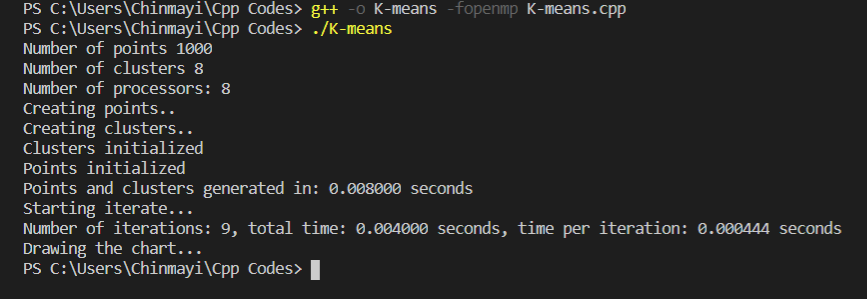
**Outputs :**

****

K = 2

****

K = 4

****

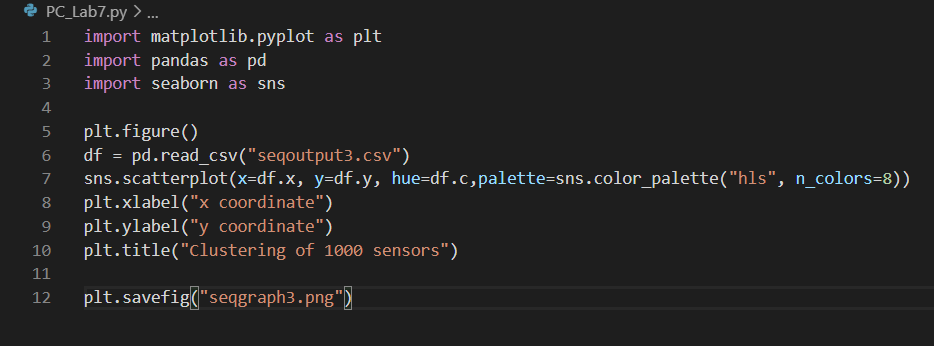
K =8

Different sections have been used to allot the section to threads without encountering race conditions.

The two for loops have been declared as two sections.

Schedule(static) has been used to compute distances to allot fixed number of chunks to the threads. This ensures that each thread gets some amount of task and they can work efficiently.

For greater number of clustering, parallel execution takes lesser time than sequential execution.



For plotting the graph