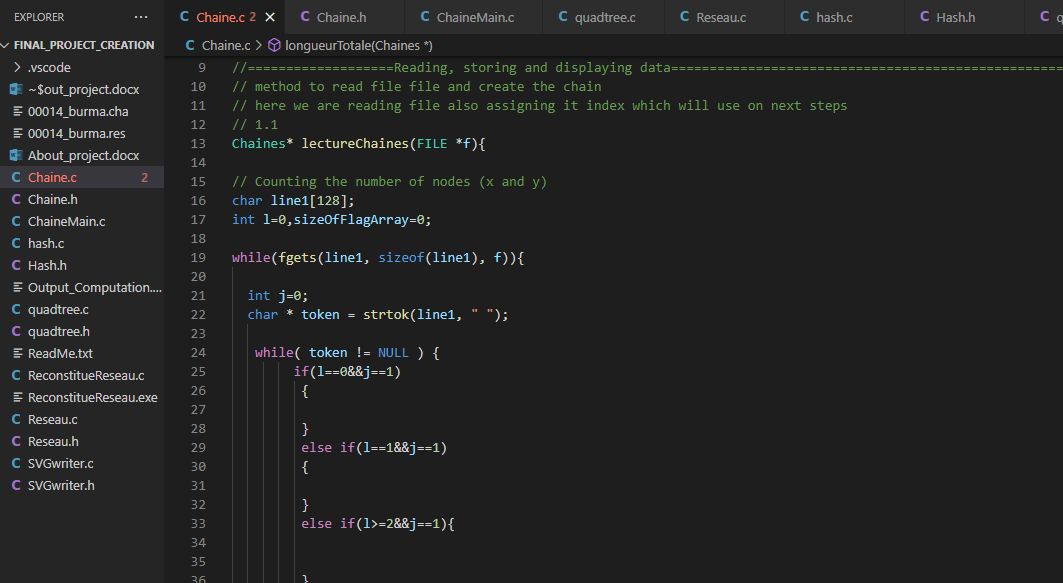
1.1:-

Q 1.1 In a Chaine.c file, implement a function Chaines\* lectureChaines (FILE \* f); who Allows to allocate, fill and return an instance of our structure from a file.

Method written in Chaine.c

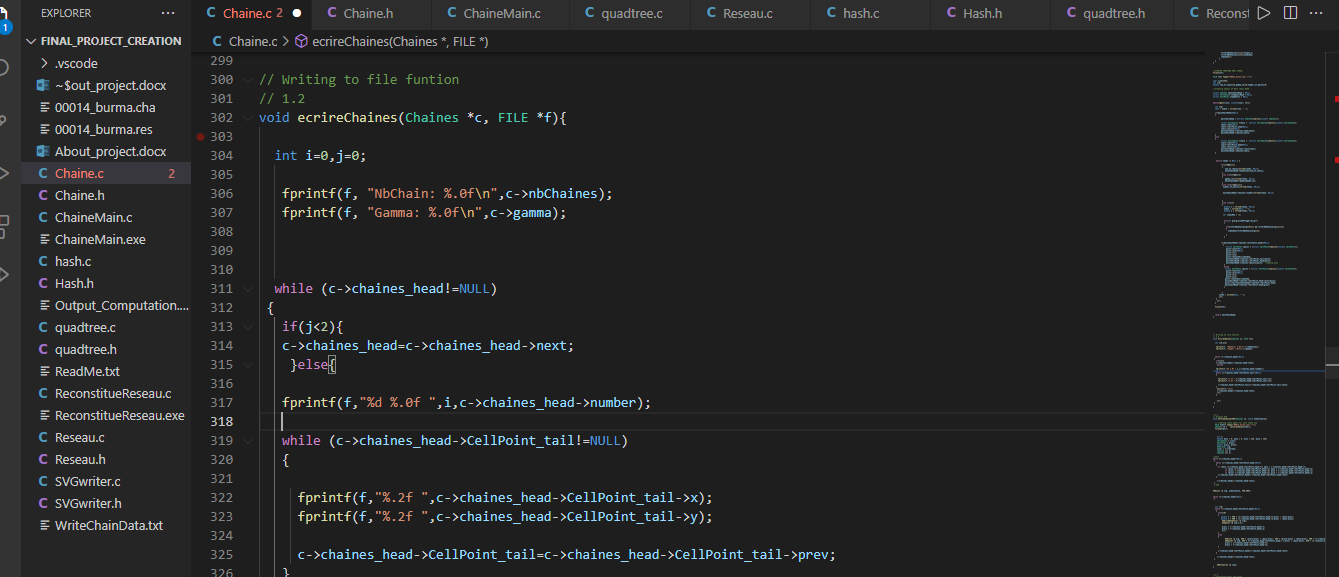


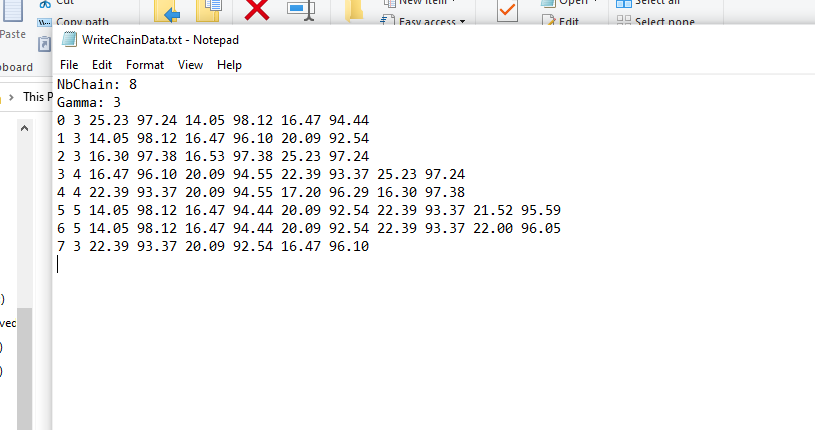
1.2

Method written in Chaine.c

Method call in ChaineMain.c

Working: after creating write that chaine data on file





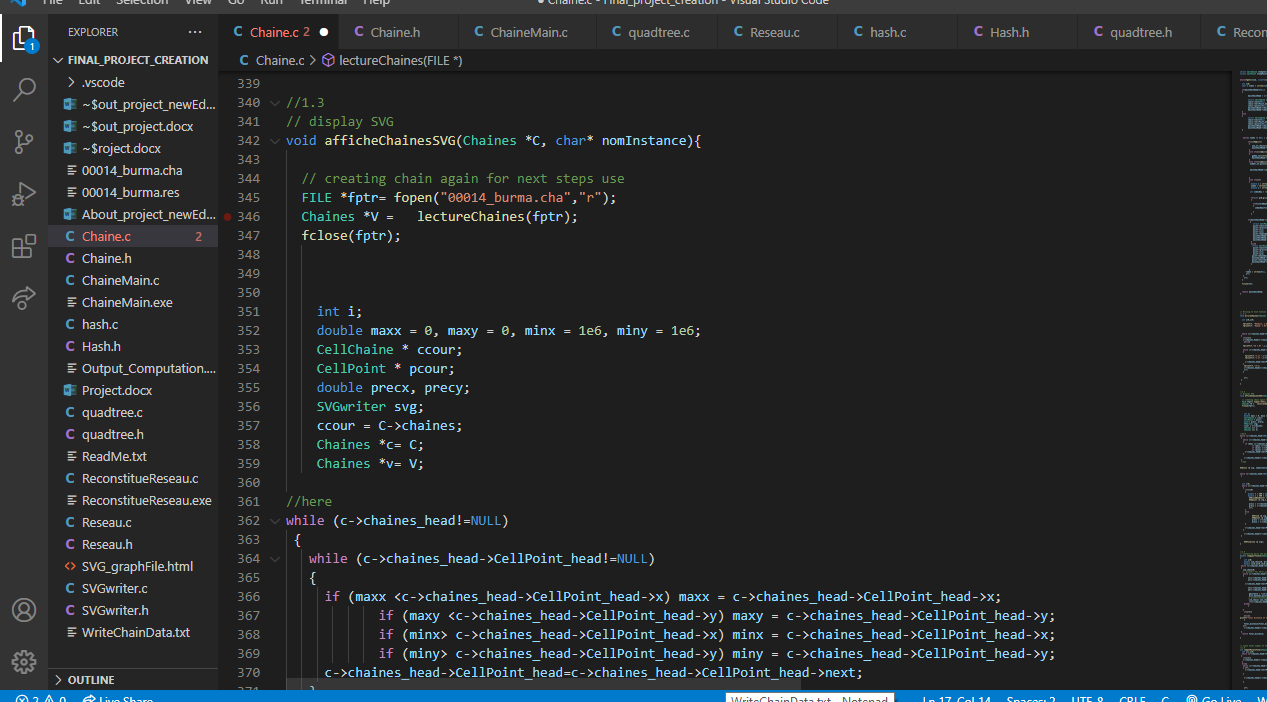
1.3

Graphical representation of instances

Method written in : Chaine.c

Method Called in ChaineMain.c

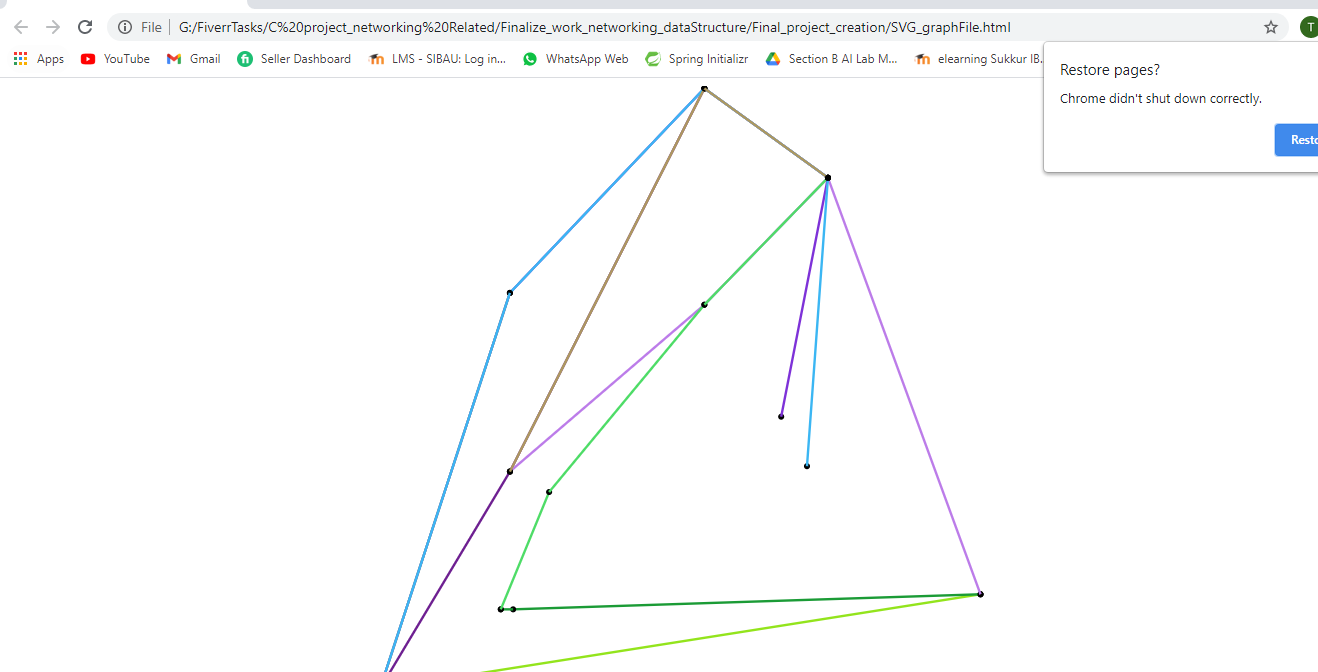
Method working: creating SVG that is showing Chaines Graphically



Enter the name of file on cmd to generate SVG on that html file



Output file:

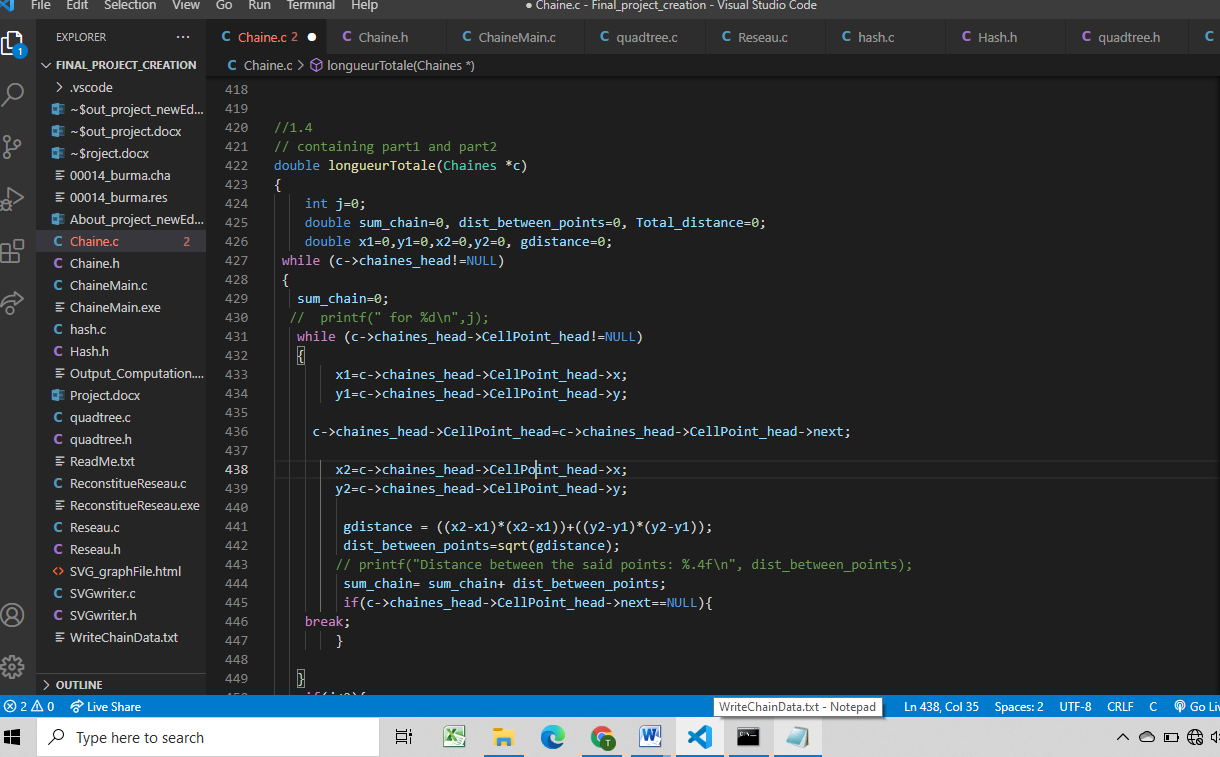


1.4 );

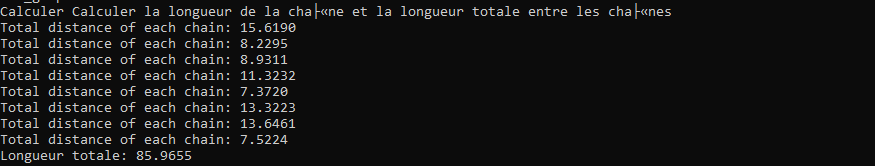
Method written in Chaine.c

Method call in ChaineMain.c

Working: which calculates the physical length of a string



Output:

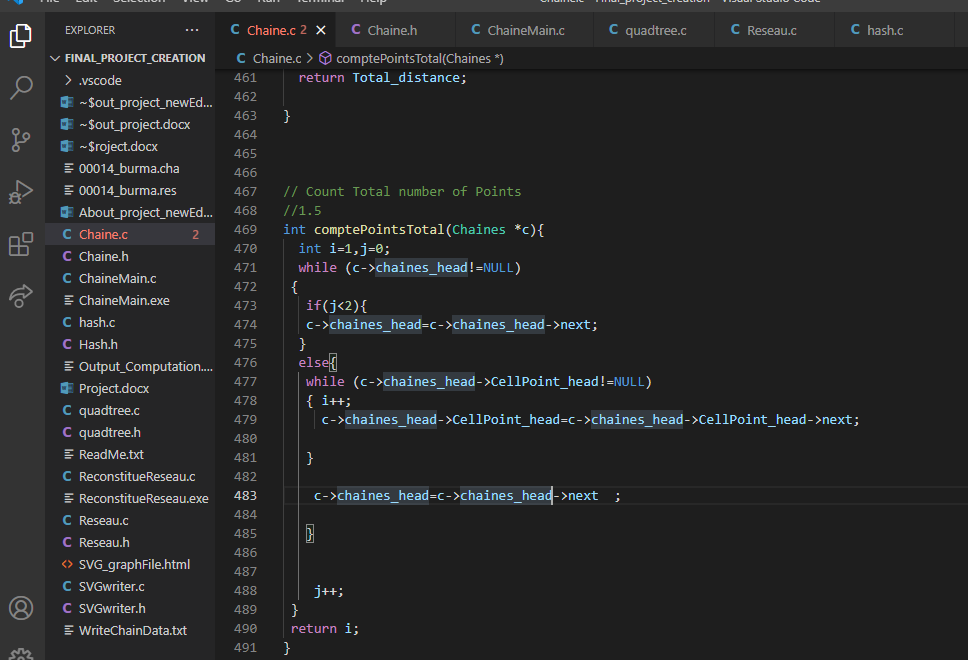


1.5

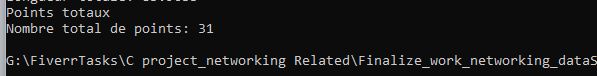
Method written in Chaine.c

Method call in ChaineMain.c

Working: which gives the total number of occurrences of points:



Output:



Network reconstitution

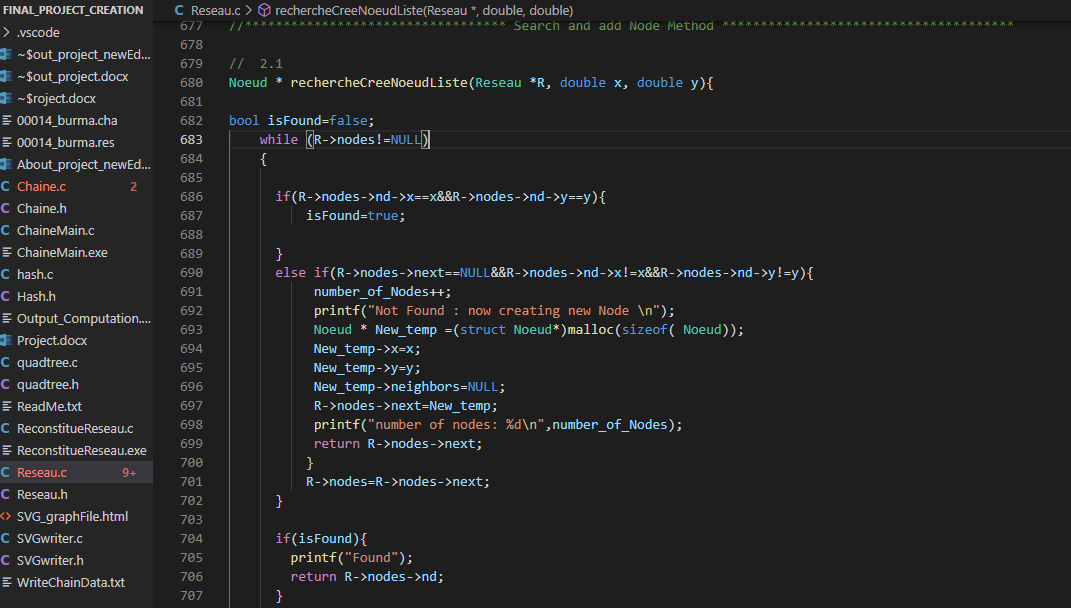
First method: storage by chained list

2.1:

Method written in Reseau.c

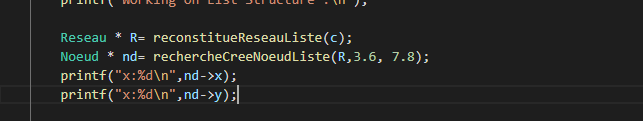
Working: which returns a Node of the network R corresponding to the point (x, y) in the list chained nodes of R.

Note that if this point exists in nodes, the function returns a node existing in nodes and that,otherwise, the function creates a node and adds it to the list of nodes in R's network. number of a new node is simply chosen by taking the number nbNodes + 1 (just before update to the value nbNodes).



Output after Calling

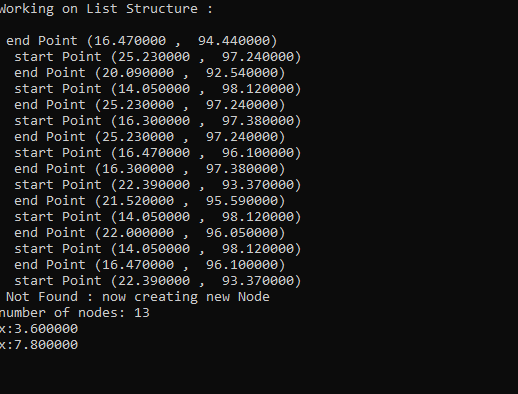
Calling this method in ReconstitueReseau.c



Output:

Given number is not part of the chain

So it will add this node in network

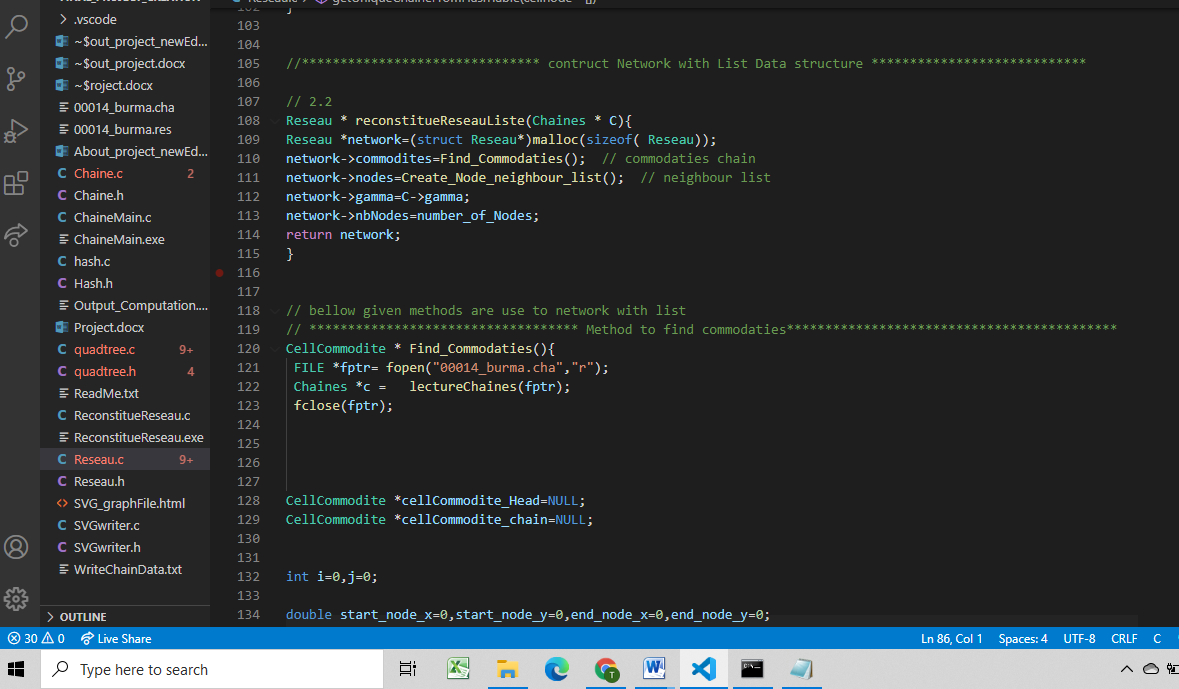


2.2

Method written in Reseau.c

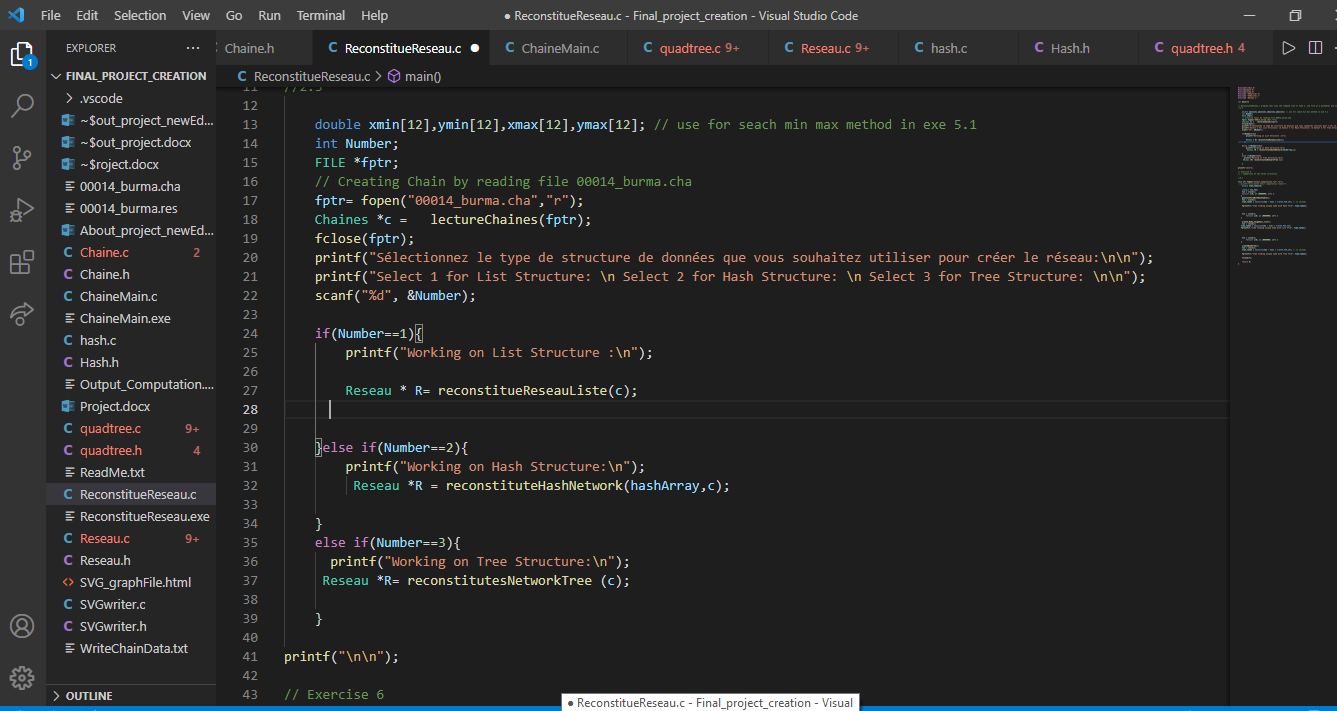
Working: reconstruct the network with list

This method is used in other exercise parts as well



2.3

Q 2.3 Create a main ReconstitueReseau.c program that uses the command line to take a .cha file as a parameter and an integer indicating which method one wishes to use (list, hash table, or tree)

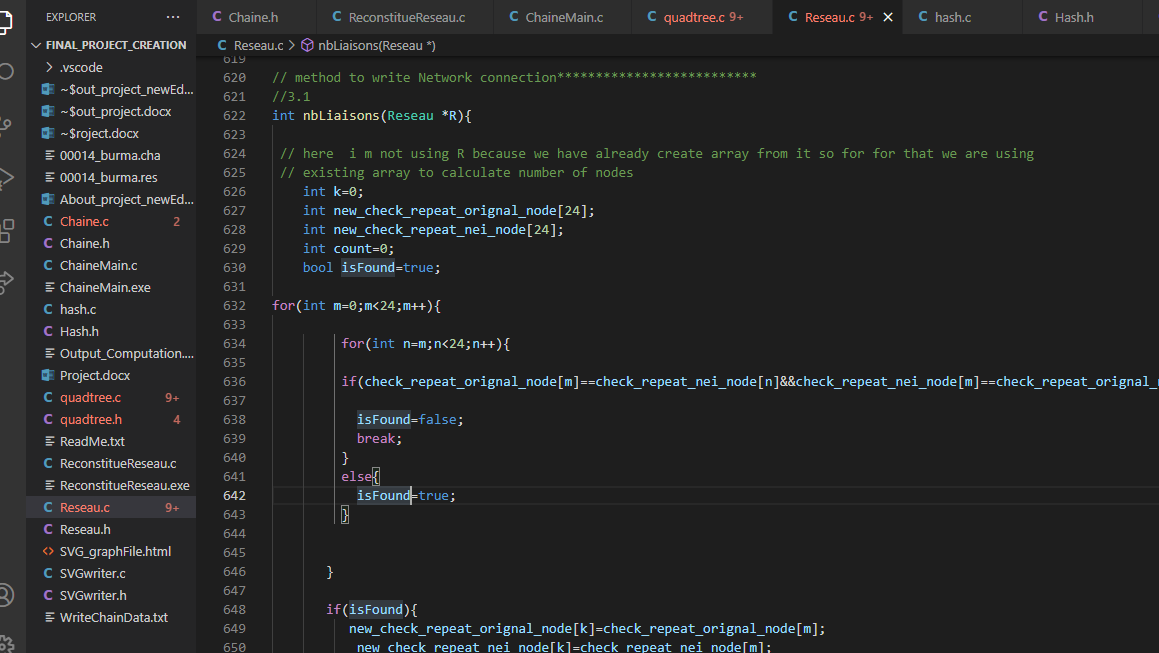


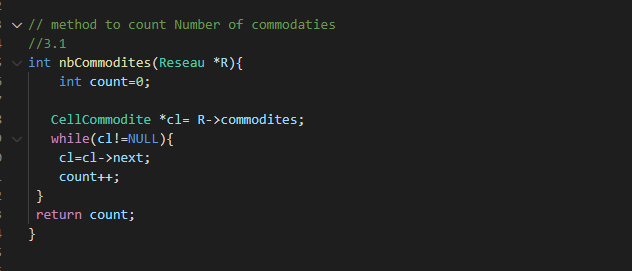
3.1-

Two methods:

Written in: Reseau.c

Working: count number of commodities and number of connection in a network

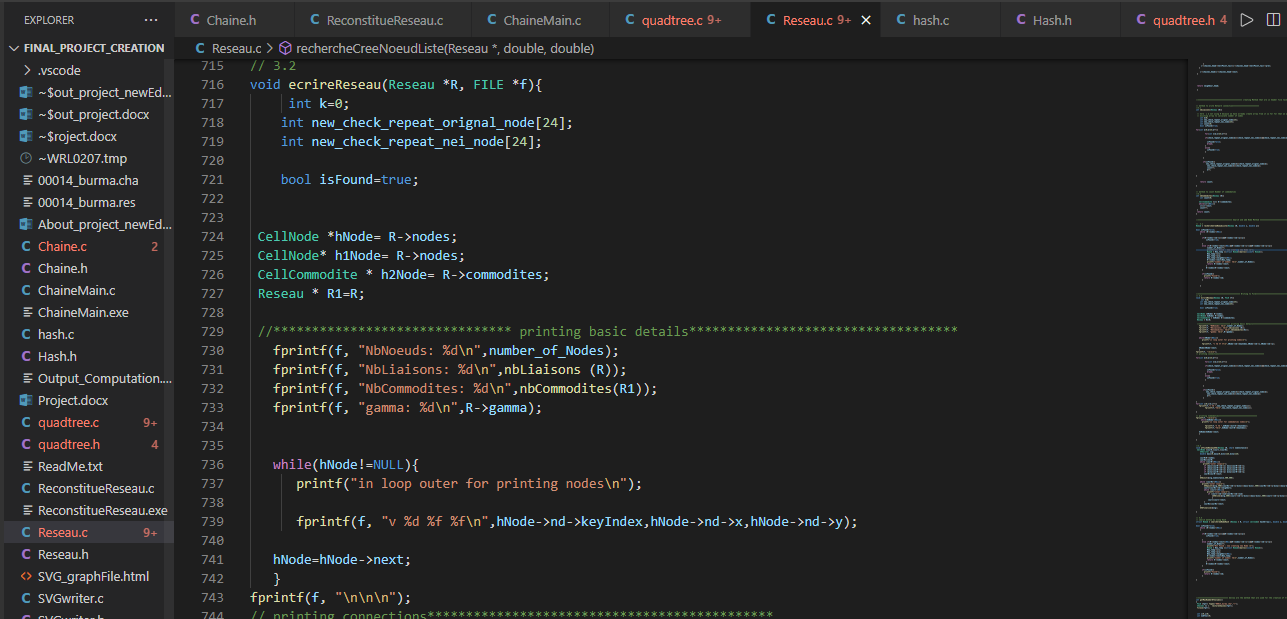




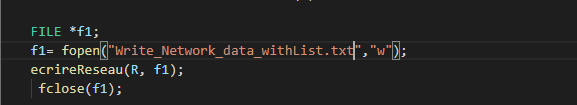
3.2:

Written in: Reseau.c

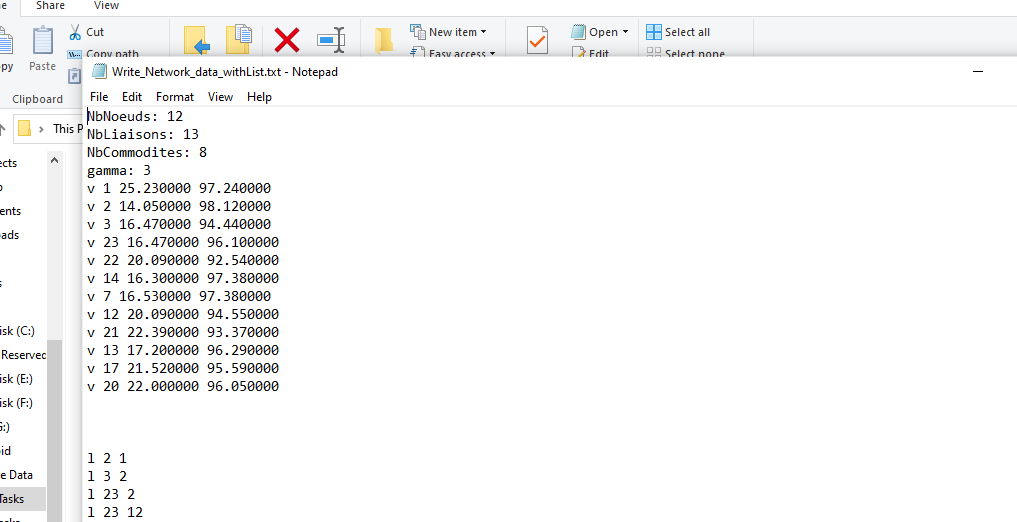
Implement a void writeReseau function (Network \* R, FILE \* f); who writes in a fi shit the contents of a Network respecting the same format of the 00014 burma.res file.



Output:



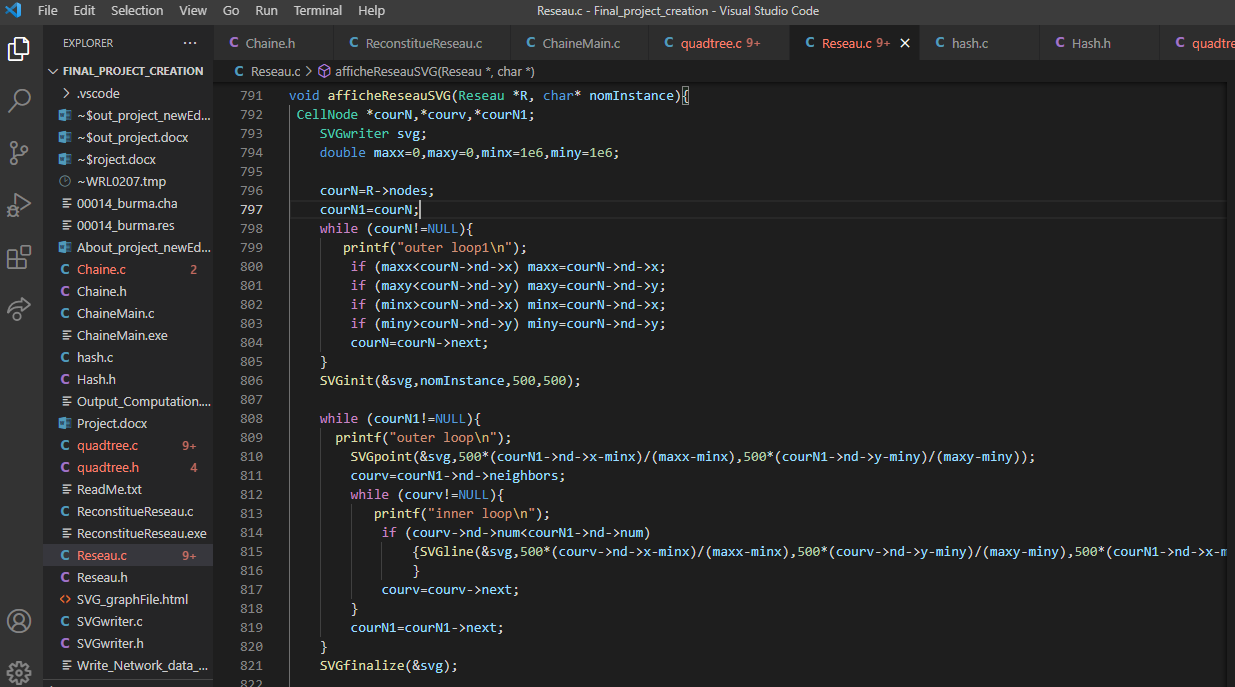
Generated file



3.3

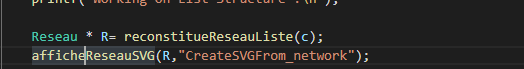
Creating SVG from list network

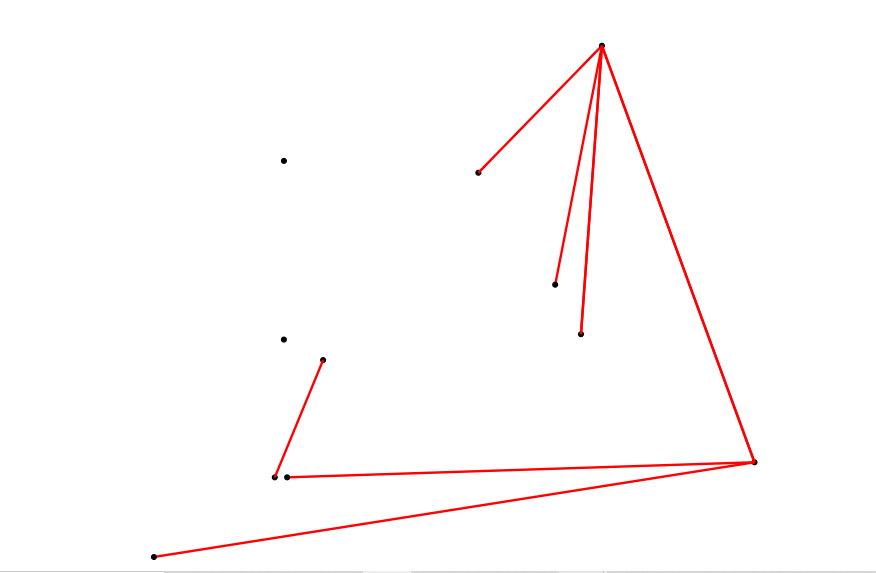
Written in : Reseau.c



Output:

Calling in ReconstitueReseau.c

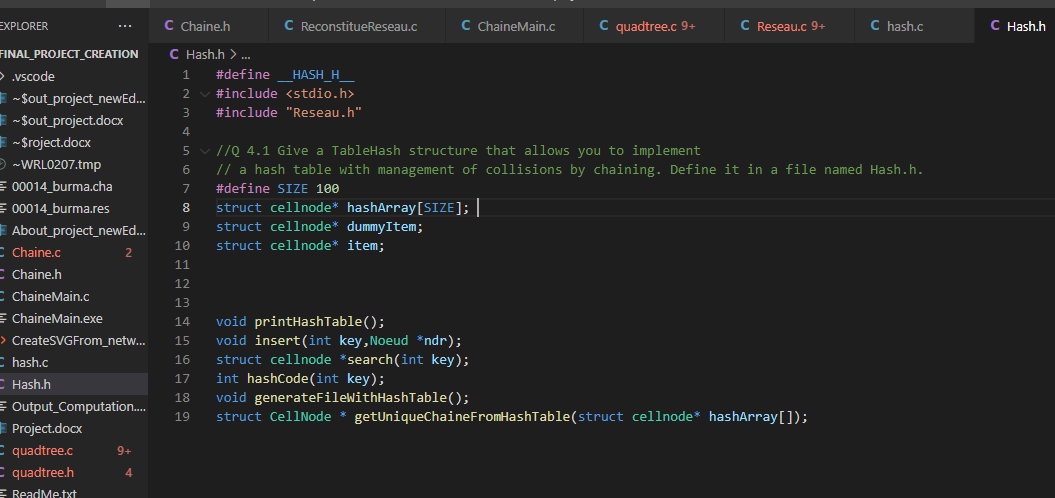




4.1

Creating structure for hash

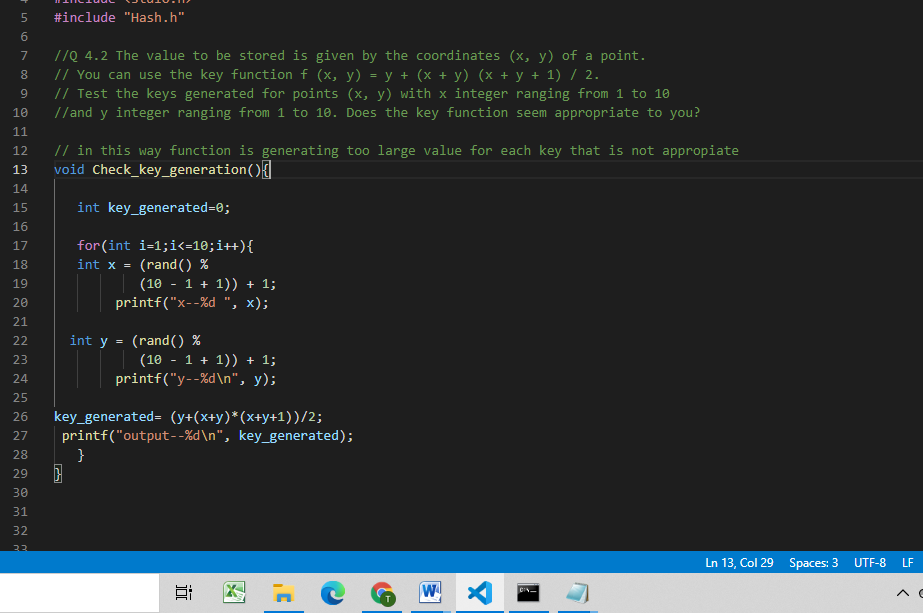
In hash.h



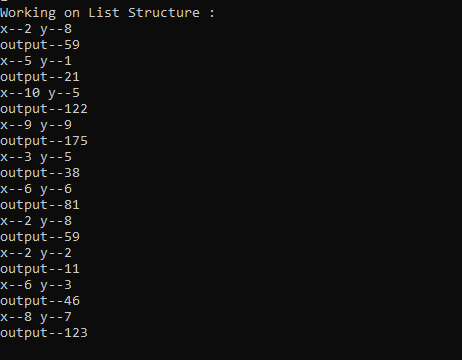
4.2:

Written in : hash.c

The value to be stored is given by the coordinates (x, y) of a point. You can use the key function f (x, y) = y + (x + y) (x + y + 1) / 2. Test the keys generated for points (x, y) with x integer ranging from 1 to 10 and y integer ranging from 1 to 10. Does the key function seem appropriate to you?

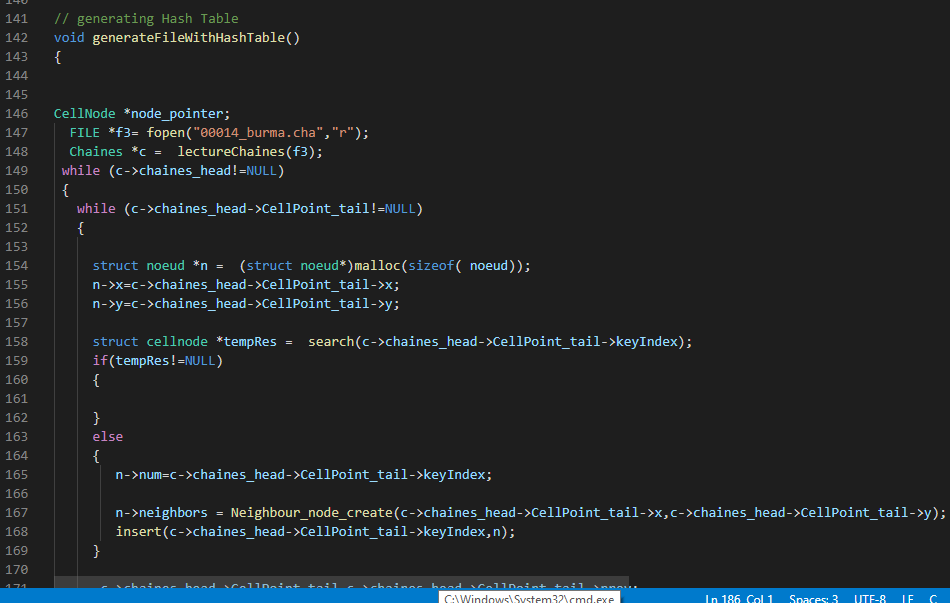


Output:



4.3:

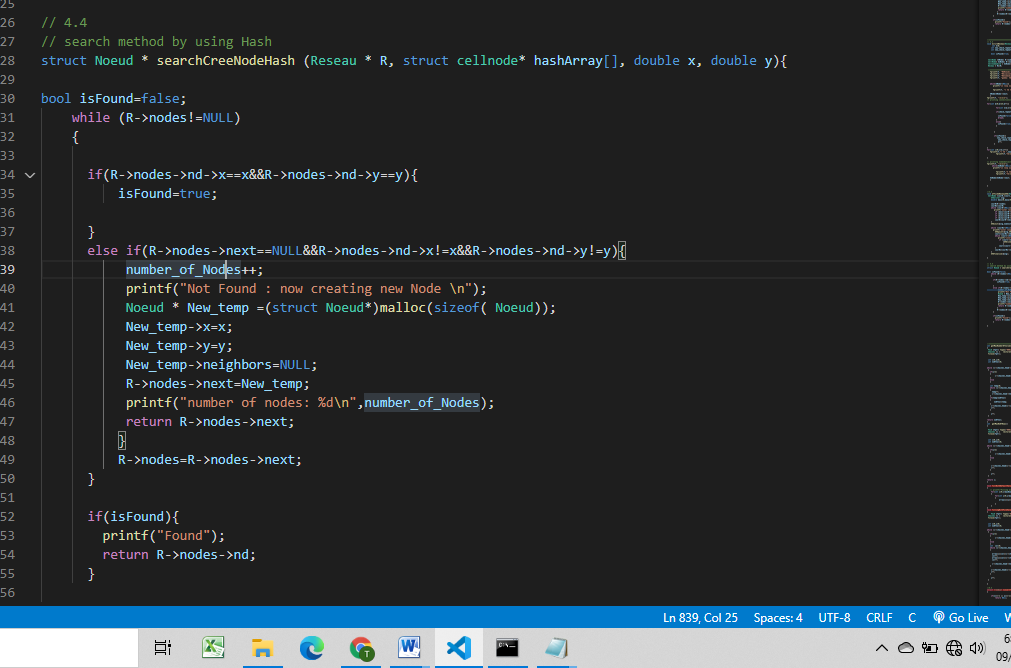
Written in hash.c



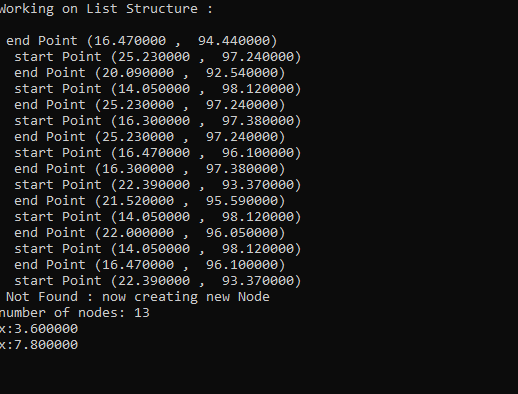
4.4

Written in Reseau.c

Working: search given node in network and hash table



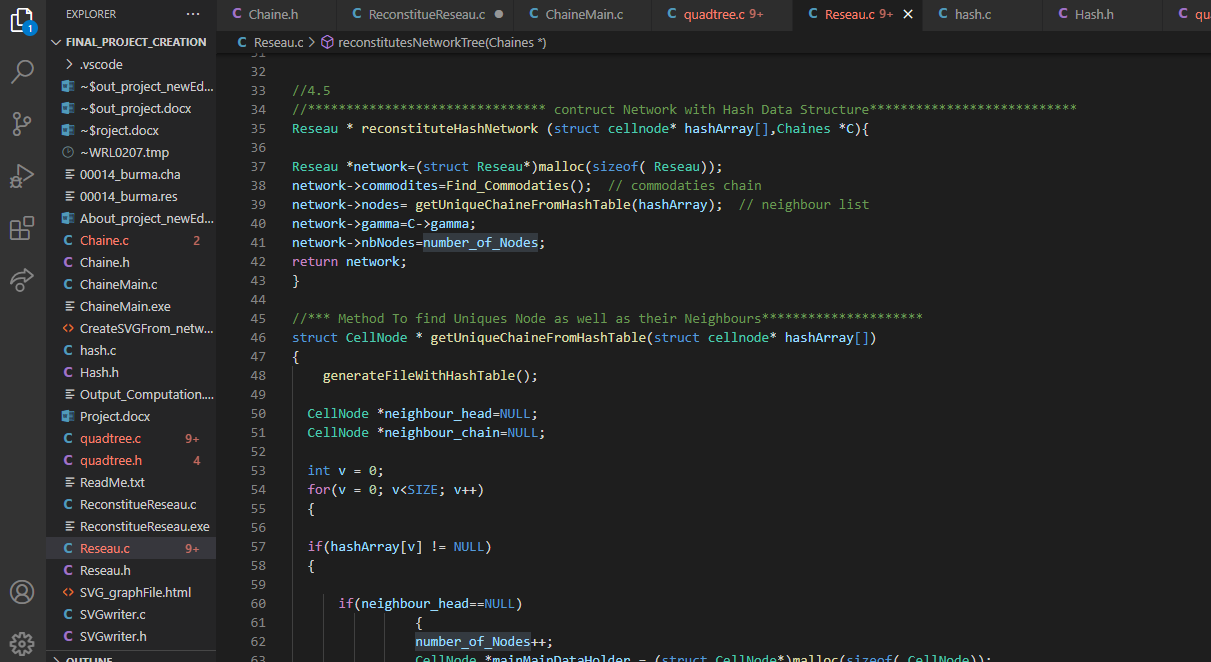
Output:



4.5

Written in Reseau.c

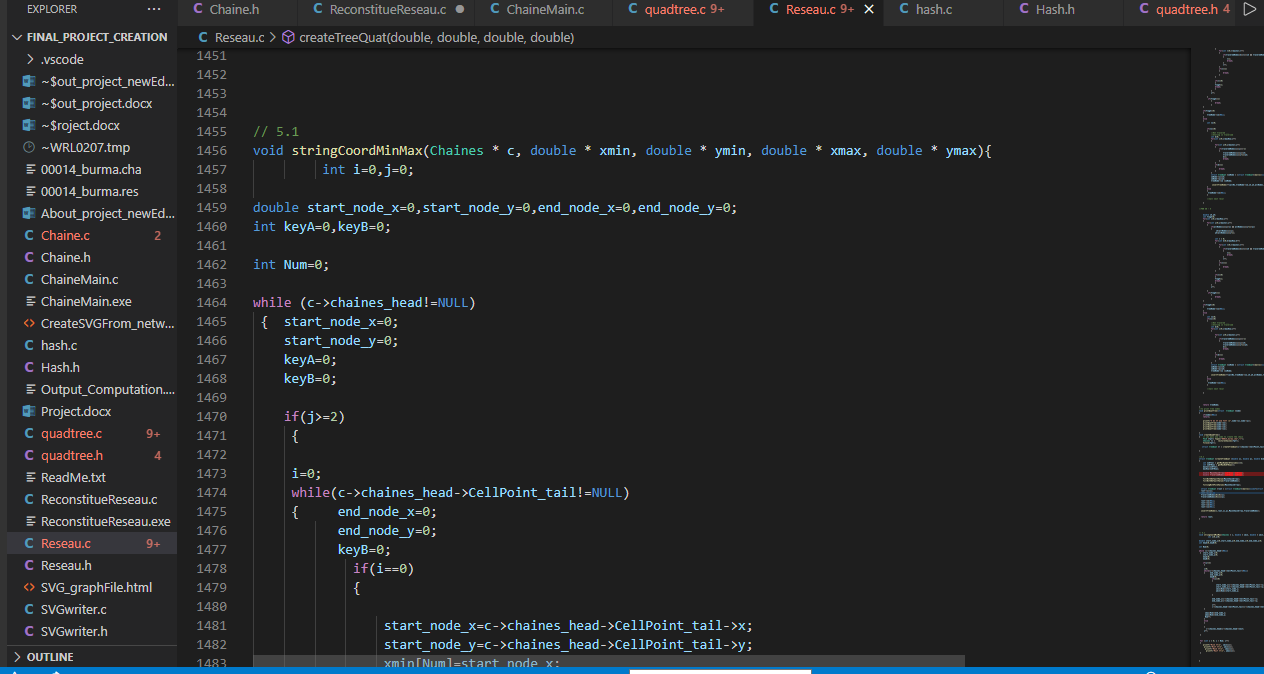
Reconstitute network with hash Table:



5.1:

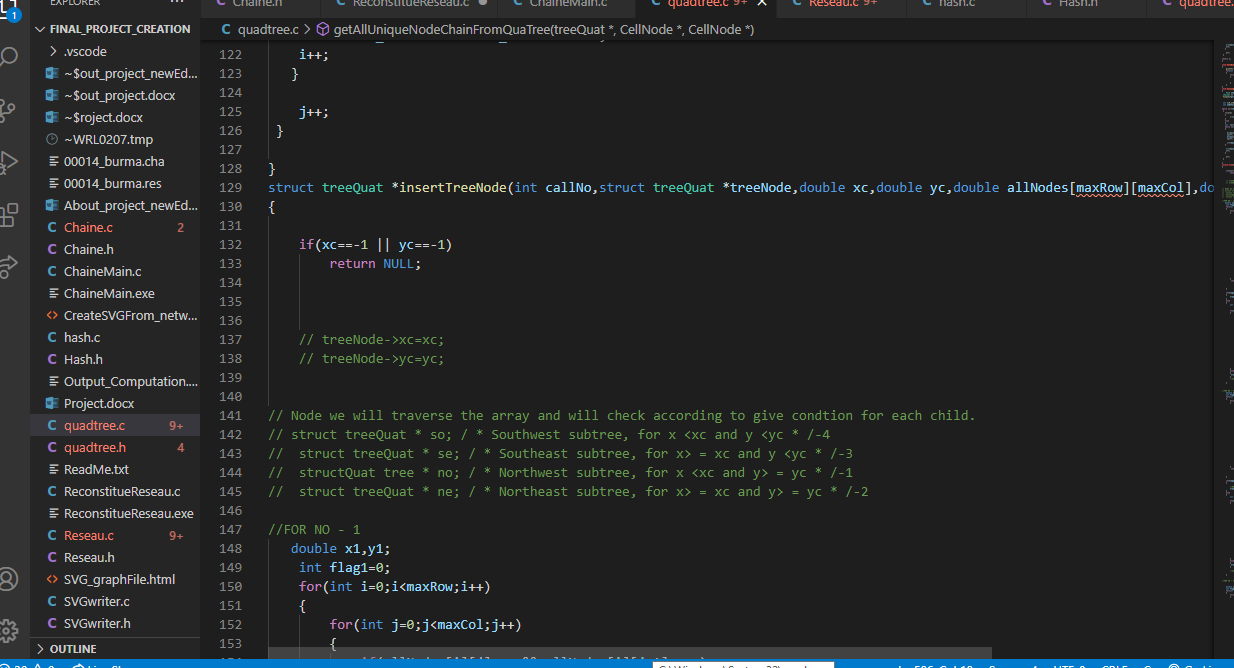
Written in quadtree.c

Working : find min and max co-ordinates



5.2

Written in quadtree.c



5.3

Written in quadtree.c

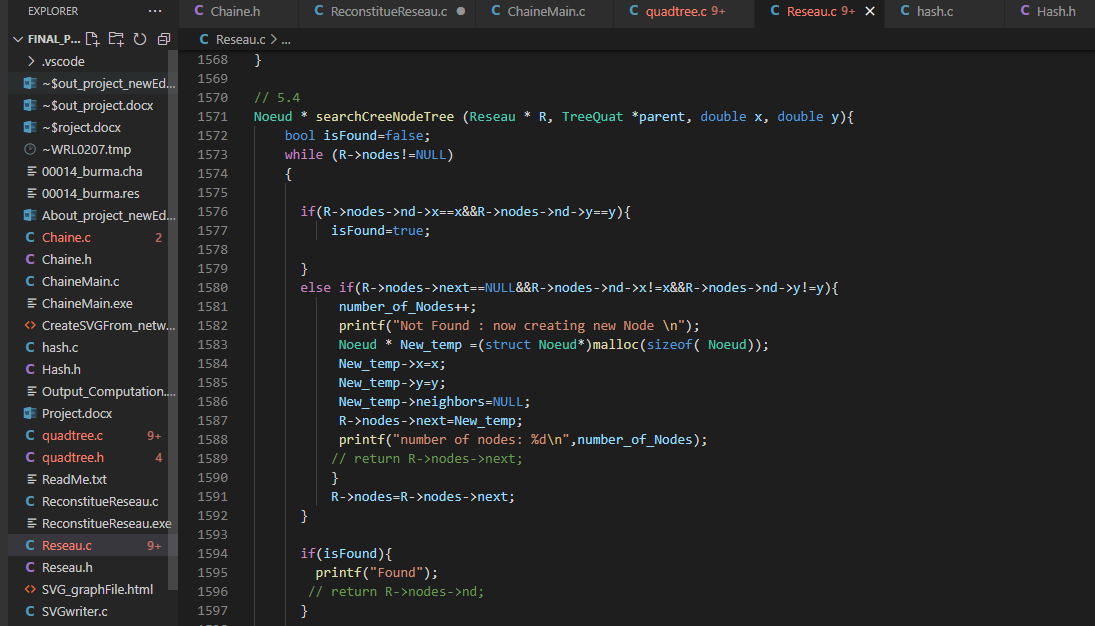
Working: creation of tree



5.4

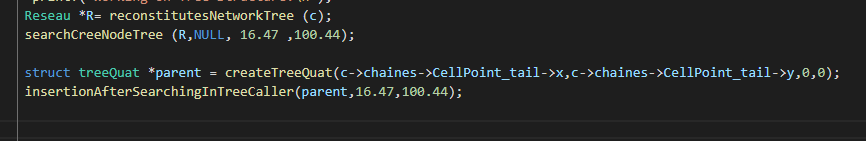
Written in Reseau.c

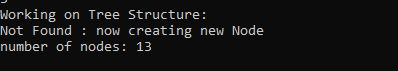
Working to search node in tree if not found than add the given node in tree as well as network





Calling:

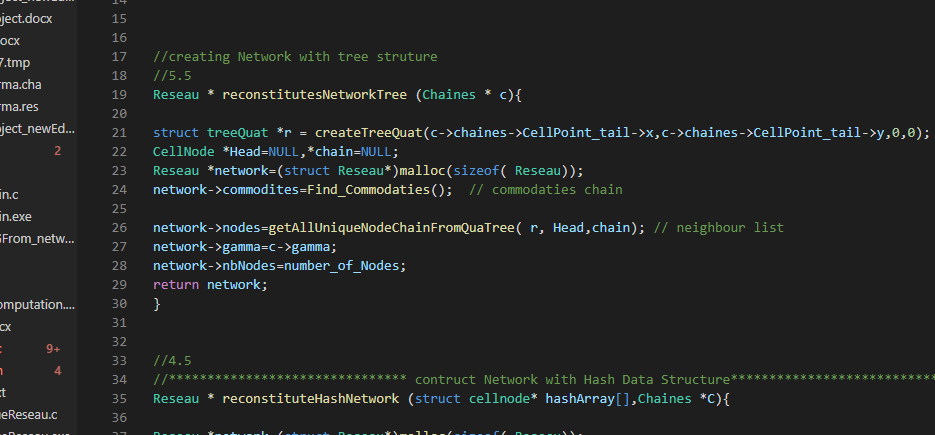




5.5

Written in Reseau.c

Working: creation of network with tree structure



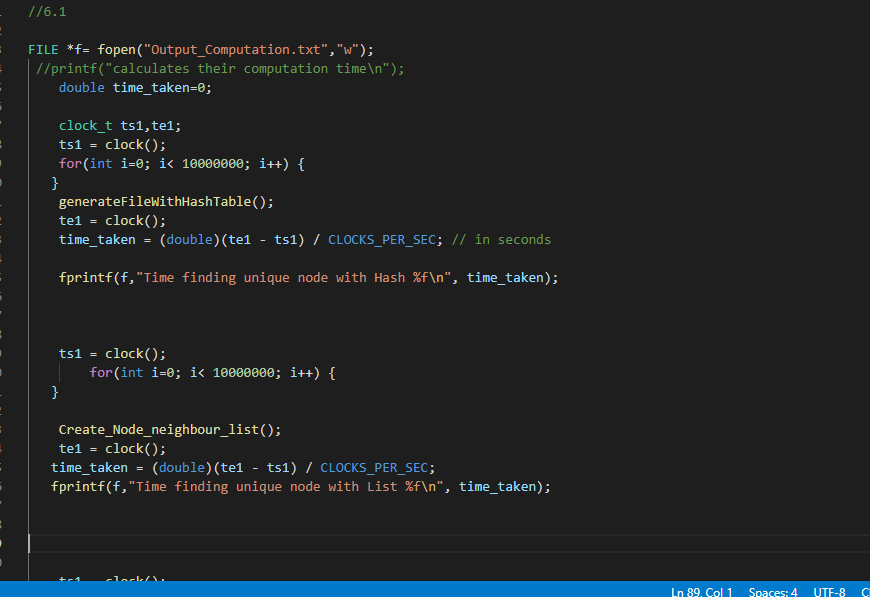
6.1:

this is computation part:

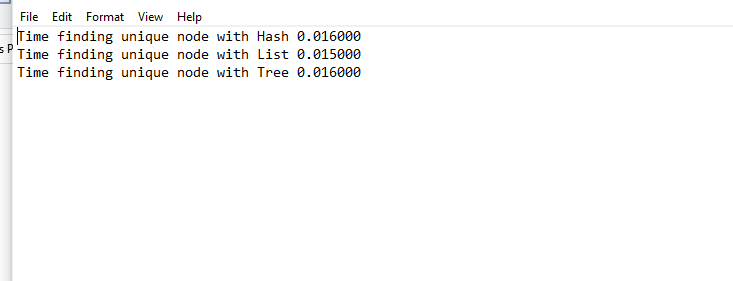
which calculate working of three structure list,hash and tree

computation time also depends on your Computer speed as well.

It is possible that it may show you 0 as a result



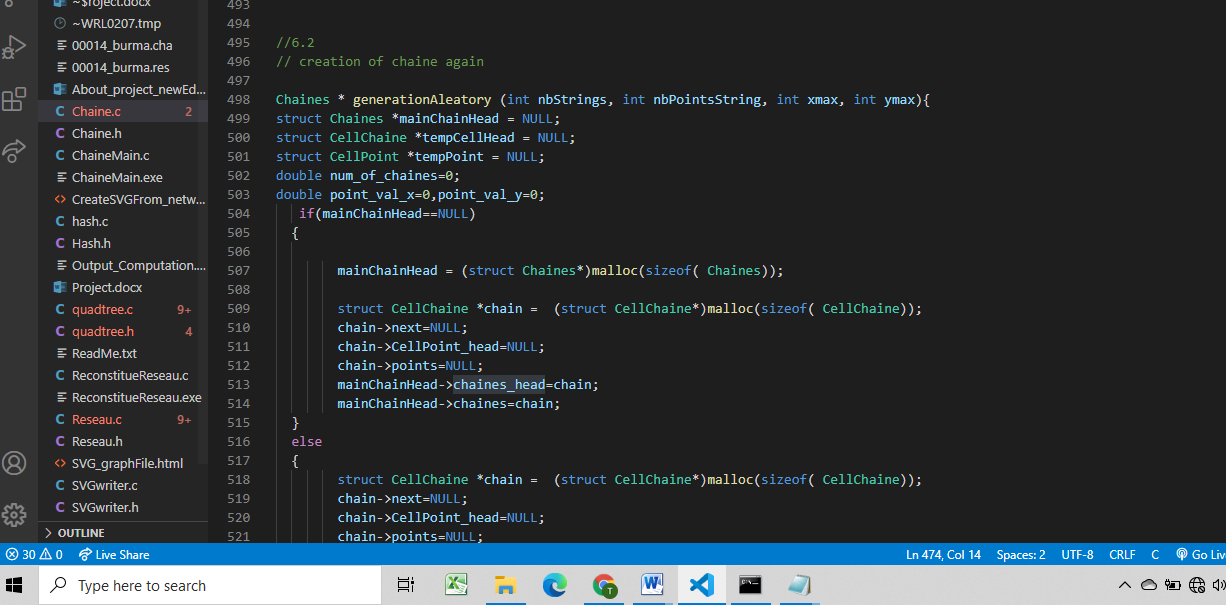
Output:



6.2:

Written in Chaine.c

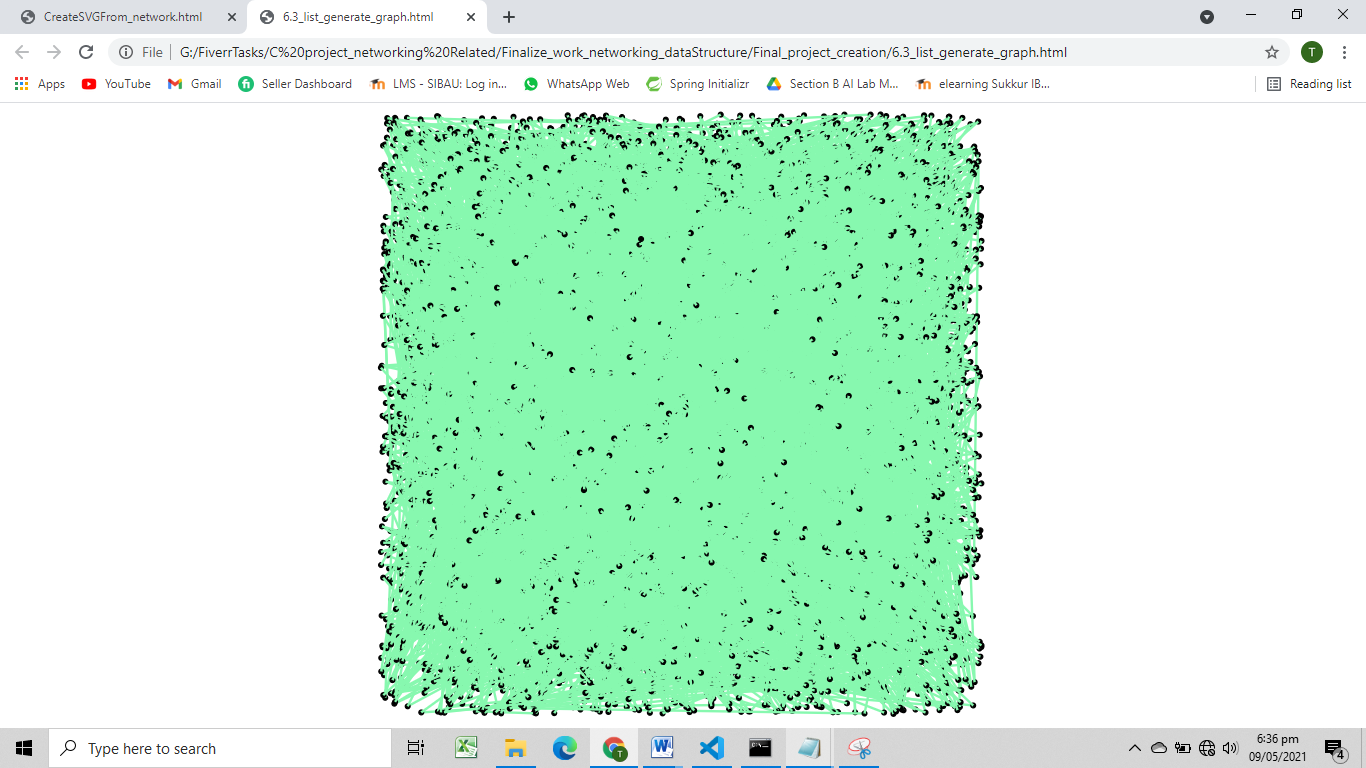
Working create a new chaine



6.3

Generation of graph

This is image of graph generating from 500 number of chaines 5000 number of points per chaine and 5000 maximum x and y



6.4

Q 6.4 Analyze your results.

Now if we analyze our created graph and generated results then we find following key points to note.

• When we changed the number of points, it created difference in processing time (Traversing time).

o For example in case of hash table it took less time than tree, and in case of tree it took less time than the time taken by list. So it can be concluded that Hash table is much faster and can play good role in creation of re construction of network then using the other datas tructrues.

• Another point to notice was the difference in generation time of different data structure.

o For example in case of hash table, it took very less time in creation then other structures.

o Other thing is that tree took much time because of recursive nature of function (Note we have created tree with the use of recursion and list by use of while loop).

• By using the conclusion of above stated points we noticed that when we create the SVG graph with usage of Hash it took less time than creating with list or tree.