On this homework we focus on json data about information from Computer Scientists networks. First of all we read data and we obtain a huge list where each element is a dictionary for example of this structure:

```
{'authors': [{'author': 'pierre seimandi', 'author_id': 1},
  {'author': 'guillaume dufour', 'author_id': 2},
  {'author': 'fran&ccedil;ois rogier', 'author_id': 3}],
 'id_conference': 'conf/iccs/2010',
 'id_conference_int': 1,
 'id_publication': 'journals/procedia/seimandidr10',
 'id_publication_int': 1,
 'title': 'a two scale model of air corona discharges.'}
```

So the key of our data is the “id_publication” and “id_publication_int” that describe the publication event in a certain conference (“id_conference” and “id_conference_int”), with a title that should be a key too (but to prevent any mistake we do not consier it as a key, in case it could be two publication with the same title) and with a list of authors that take part to the publication, specified with their name (“author”) and their identification (“author_id”).
 Then we create a dictionary where each key is “an author_id” and each value is its list of publication, in this way:

```
{1: [1],
 2: [1, 783, 56682, 56818],
 3: [1, 783],
 4: [2,
  1489,
  1782,
  171421,
  171578,
  513900,
  514004,
  748166,
  748215,
  830766,
  831936,
  931068],
 5: [2, 171423, 524391, 748826],
 6: [3, 163784],
.
.
.
```

Now we create a graph where each node is an author_id and we use the dictionary that we have built in the previous step to calculate linkage between nodes; infact using the funtion of Jacard Distance we create a link between two nodes if and only if this two authors have at least one common publication and we attribute a weight equal to their Jacard distance to their link.
After we write an id_conference_int in input and we are going to build a subgraph where there will be only nodes of authors that take part in the input conference; so not all nodes are conntected beacuse if two connected authors in the graph are not both in the input conference but only one of them, then the linkage will not be of course in the subgraph. Instead if two authors are connected in the graph and they both take part in the input conference, then they are connected in the subgraph too. I calculate some Degree measure of the subgraph: I obtain that the author_id 148131 has the

highest value of degree equal to 48, for the given input conference 16501; then for a given input author I obtain its value of Closeness Centrality, that is the answer of the question “thus the more central a node is, the closer it is to all othe nodes”. Then we write an author_id and a pivot value in input to obtain another subgraph from the original graph where there are all nodes that have hop distance at most equal to the input pivot with the input author.

In the last exercise, using the meaning of Erdos Number, I want to return the weight of the shortest path that connects an input author with Aris; to do it I create a function “sp(G,a1,a2)” that returns the weight of the shortest path between authors a1 and a2, using the Dijkstra algorithm and where each weight is always the Jacard distance between authors. Given in input the author_id 519479, I obtain 3.82 “path weight” to connect to Aris, where this value is the sum of the edge that we visited to go from the input author to Aris. In the end I write a last Python software to obtain the Group Number of all nodes “v” in the graph, given a subset of nodes “I”.

$$GroupNumber(v) = min\{ShortestPath(v,u)\}$$

The Group Number of a node is the path that visited the less number of nodes between all shortest paths from the node “v” of the graph to the node “u” in the subset “I”.