Input: $\mathbf{D}, \mathbf{E}$ ▷ Datasets and datasets duplicated
Output: $\mathbf{C}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$ ▷ Four tsv files containing the matches(datasets, classes and properties).

1: $\mathbf{M}_e\{\}$ ▷ HashMap containing datasetTarget and another hash containing properties and classes exact matched
2: $\mathbf{M}_s\{\}$ ▷ HashMap containing datasetTarget and another hash containing properties and classes matched using String similarity
3: $\mathbf{M}_a\{\}$ ▷ HashMap containing datasets already compared
4: $\mathbf{D} - \mathbf{E}$ ▷ Remove duplicates contained in $\mathbf{E}$
5: $\mathbf{T} = \mathbf{D}$
6: **for all** $\mathbf{d} \in \mathbf{D}$ **do** ▷ In parallel
7: **for all** $\mathbf{t} \in \mathbf{T}$ **do**
8: **if** $(\mathbf{d} <> \mathbf{t}) \wedge$ **alreadyCompared**$(\mathbf{d}, \mathbf{t})$ **then**
9: continue ▷ Skip the current $\mathbf{d}$ and $\mathbf{t}$
10: **end if**
11: $\mathbf{M}_e$.add(**getExactMatches**$(\mathbf{d}, \mathbf{t})$) ▷ Add a set containing the properties/classes that are exact the same in both datasets
12: $\mathbf{M}_s$.add(**getSimMatches**$(\mathbf{d}, \mathbf{t}, 0.8, \mathbf{M}_e)$) ▷ Add a set containing the properties that are similar in both datasets, excluding the matches from $\mathbf{M}_e$
13: $\mathbf{M}_a$.add(d, t) ▷ Add the dataset pair already processed
14: **end for**
15: **end for**
16: **printMaps**$(\mathbf{M}_e, \mathbf{M}_s)$ ▷ Print the content of $\mathbf{M}_e$ and $\mathbf{M}_s$ inside the files $\mathbf{C}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$