Task 3 (10%). Compression over Columnar Representation

Consider the columnar representation that you constructed in Task 2. In this task you are asked to update your Task 2 implementation and apply dictionary-based compression in order to efficiently process queries with equality predicates. Apart from the previous inputs, your program will take as additional input the value *compressedCol* which denotes the column that you need to compress. Update accordingly the processing of the query in order to exploit the usage of the dictionary.

Output: The format of the result must be CSV (separated by ‘,’).

Data: "lineitem.csv" (download here), CSV format where the fields of each tuple are separated by ','.

Deliverables:

- CompressedColumnStore.java, the program should receive six arguments:
  - <input>: denoting the input dataset.
  - <output>: denoting the output file.
  - <schema>: a string containing the schema of the dataset (e.g., “attr1:String,attr2:Int”).
  - <projectionList>: a string containing the arguments projected (e.g., “attr1,attr3”).
  - <whereList>: a string containing the where clause (e.g., “attr1|=|3,attr2|<|7”).
  - compressedCol: a string referring to the to-be compressed column.