In order to do a naïve data linkage without any blocking, each product in the amazon_small.csv file was compared with each product in the google_small.csv file. First of all, the fuzzywuzzy method of comparison was chosen in order to compare each whole title string with the whole title strings in the other data file. The fuzzywuzzy function fuzz.token_set_ratio compares two strings by the words in the string, irrespective of their position, and returns an integer based on how similar the two strings are considered to be on a scale of 1 to 100. Hence, this function was fit to compare the two data sets by title, which only contain a small amount of words. For each title in one data set, it was compared with every title in the other data set and the highest scoring pairs were included in a list. However, this may produce highest pairs where one data in google_small.csv is paired with multiple datas in amazon_small.csv. Therefore, this list was tidied into a dictionary with idGooglebase as the key and one or possibly multiple pairs as the values. From this dictionary, for each key, the pair with the highest fuzzywuzzy return value was appended to the final list which was to be produced as the output task1a.csv file. Hence, each data was matched with another data set in this way.

To increase the precision of the matches, the element 'Price' could have also be used along with the 'title'. However, comparing the titles seemed sufficient when comparing the two data sets as the precision and recall was over 0.9. Also, this way of comparison is only valid for these data sets and specifically the titles of the data sets. It cannot necessarily be used to compare other kinds of datas, unless they also have short strings.

For the blocking method in part 1b, every word in the titles of the product data was used to create a dictionary consisting of each word as a key and the frequencies of the words as the value. The words that appeared more than once in the titles were made as block keys which was then used to classify and place the products of each data set into blocks. The condition of more than once was so the block keys unique to the title is excluded, which would mean that is simply comparing the two data sets as a whole.

However, this method may produce too many blocks. Hence, the amount of time to execute the program is much close to the execution time of comparing each data set and therefore is inefficient. In order to improve this, a more generalized key should be more appropriate unlike this method which focuses on making keys based on the amazon data set.