Name: Sai Anish Garapati
UIN: 650208577

Assignment 2:

Functionality of the code:

Task 1:

- After reading the documents, each document is processed separately to form the terms that would be used for indexing in this way:
    - SGML tags are removed using regular expressions
    - All numbers are removed using regular expressions
    - **word_tokenize()** from nltk.tokenize is used to split words on white spaces
    - All punctuations are removed using **string.punctuation** and words are converted to lowercase
    - Stop words are removed from the list based on **stopwords set from nltk.corpus**
    - **PorterStemmer** from **nltk.stem** is used to stem the words and stop words from the resultant list are removed
    - Words with one or two characters are removed

- From the processed list of docs which contains lists of terms from each doc, Inverted index table is built using a **dictionary** from python with **index terms as keys and dictionaries (with document ids in which term has occurred as keys and term frequencies as value) as values. Document frequency of a term is obtained using the length of the dictionary.**
- Document ids start from zero for convenience in this list.
  Ex: Entry in inverted index table for a term 'experiment':
  {'experiment': {0: 2, 10: 1, 11: 1, 15: 1, 16: 1, 18: 1, 24: 1, 28: 1, 29: 2, 34: 1, 36: 1, 40: 1, 42: 1, 46: 1, 51: 1, 52: 1, 57: 1, 68: 1, 69: 1, 73: 1, 77: 2, 83: 2, 98: 2, 100: 1, 102: 1, 111: 1, 114: 1, 120: 1, 122: 3, 130: 1, 136: 1, 139: 1, 141: 1, 153: 1, 155: 1, 166: 1, 167: 1, 169: 1, 170: 2, 172: 2, 175: 1, 178: 2, 182: 1, 183: 1, 185: 3, 186: 1, 187: 1, 188: 1, 190: 1, 194: 3, 196: 2, 201: 1, 202: 1, 205: 2, 206: 2, 211: 1, 215: 1, 219: 1, 221: 1, 224: 2, 226: 1, 229: 1, 233: 4, 244: 1, 250: 1, 255: 2, 256: 1, 261: 1, 270: 2, 272: 1, 276: 1, 281: 1, 282: 1, 285: 1, 286: 1, 288: 1, 293: 1, 294: 1, 303: 1, 306: 1, 328: 2, 329: 2, 333: 2, 337: 1, 338: 1, 343: 2, 344: 1, 345: 3, 346: 1, 353: 1, 359: 1, 368: 1, 369: 1, 371: 2, 376: 1, 396: 1, 408: 1, 410: 2, 412: 2, 417: 1, 419: 1, 420: 1, 422: 1, 426: 1, 432: 1, 434: 1, 438: 1, 440: 2, 441: 3, 442: 1, 452: 1, 454: 2, 461: 1, 463: 1, 466: 1, 483: 3, 493: 2, 495: 1, 496: 1, 497: 1, 500: 1, 502: 1, 503: 1, 504: 1, 510: 1, 516: 1, 517: 2, 518: 1, 519: 2, 521: 3, 535: 1, 539: 1, 543: 2, 548: 1, 551: 2, 552: 1, 557: 1, 562: 1, 566: 1, 568: 1, 571: 4, 575: 1, 587: 1, 594: 1, 599: 1, 605: 1, 609: 1, 631: 1, 633: 1, 634: 1, 635: 1, 643: 1, 644: 1, 648: 1, 657: 1, 661: 1, 662: 2, 665: 2, 669: 1, 674: 1, 677: 1, 678: 1, 684: 3, 687: 3, 688: 2, 693: 1, 703: 2, 711: 1, 712: 1, 716: 1, 719: 1, 724: 1, 727: 1, 728: 1, 738: 1, 739: 1, 742: 1, 752: 1, 759: 3, 763: 1, 765: 3, 766: 3, 771: 2, 780: 2, 789: 1, 800: 2, 801: 1, 805: 1, 815: 1, 819: 1, 822: 2, 824: 1, 826: 2, 829: 1, 835: 4, 843: 1, 844: 2, 845: 1, 846: 1, 855: 3, 856: 2, 857: 2, 862: 2, 865: 2, 866: 1, 868: 1, 877: 1, 880: 1, 886: 1, 890: 2, 906: 1, 910: 1, 911: 2, 922: 1, 923: 1, 926: 3, 927: 3, 931: 2, 934: 2, 945: 2, 949: 3, 950: 1, 953: 2, 954: 1, 958: 1, 960: 1, 963: 1, 964: 1, 972: 1, 973: 1, 983: 2, 985: 3, 995: 1, 996: 3, 998: 1, 1005: 1, 1007: 1, 1015: 1, 1018: 3, 1027: 1, 1038: 3, 1039: 1, 1044: 1, 1045: 1, 1048: 2, 1050: 1, 1061: 2, 1065: 4, 1068: 1, 1069: 1, 1073: 2, 1074: 2, 1075: 1, 1077: 1, 1079: 1, 1080: 1, 1081: 1, 1082: 1, 1091: 1, 1096: 2, 1097: 1, 1109: 1, 1111: 1, 1117: 2, 1121: 2, 1124: 1, 1126: 1, 1144: 1, 1145: 1, 1150: 1, 1152: 1, 1154: 1, 1155: 2, 1157: 1, 1158: 1, 1159: 2, 1160: 2, 1166: 1, 1170: 1, 1171: 1, 1176: 1, 1184: 1, 1185: 1, 1186: 1, 1191: 1, 1194: 1, 1195: 2, 1197: 1, 1198: 1, 1203: 2, 1204: 1, 1208: 3, 1211: 1, 1212: 2, 1213: 3, 1215: 1, 1217: 1, 1219: 1, 1221: 1, 1224: 3, 1226: 1, 1227: 1, 1229: 1, 1230: 1, 1233: 1, 1236: 1, 1260: 1, 1261: 1, 1262: 2, 1263: 2, 1267: 1, 1268: 2, 1276: 2, 1289: 1, 1297: 1, 1301: 2, 1309: 1, 1313: 2, 1316: 1, 1318: 1, 1323: 1, 1336: 3, 1337: 2, 1338: 1, 1340: 1, 1351: 2, 1362: 2, 1363: 1, 1368: 1, 1371: 1, 1373: 1, 1377: 1, 1383: 1, 1389: 1, 1391: 1, 1395: 1, 1396: 1},

Here, document frequency of ‘experiment’ will be len(inverted_index[‘experiment’]).

- By looping through each document, respective document lengths are calculated using the term frequency from inverted index table and document frequency (length of the dictionary corresponding to a term)

Task 2:

- Queries are read and preprocessed in the same way as the document corpus
  Queries after processing: [{'investig': 1, 'made': 1, 'wave': 1, 'system': 1, 'creat': 1, 'static': 1, 'pressur': 1, 'distribut': 1, 'liquid': 1, 'surfac': 1}, {'anyon': 1, 'investig': 1, 'effect': 1, 'shock': 1, 'gener': 1, 'vortic': 1, 'heat': 1, 'transfer': 1, 'blunt': 1, 'bodi': 1}, {'heat': 1, 'transfer': 1, 'blunt': 1, 'bodi': 1, 'absenc': 1, 'vortic': 1}, {'gener': 1, 'effect': 1, 'flow': 1, 'field': 1, 'reynold': 1, 'number': 1, 'small': 1}, {'find': 1, 'calcul': 1, 'procedur': 1, 'applic': 1, 'incompress': 1, 'laminar': 1, 'boundari': 1, 'layer': 1, 'flow': 1, 'problem': 1, 'good': 1, 'accuraci': 1, 'reason': 1, 'comput': 1, 'time': 1}, {'paper': 1, 'applic': 1, 'problem': 1, 'calcul': 1, 'procedur': 1, 'laminar': 1, 'incompress': 1, 'flow': 1, 'arbitrari': 1, 'pressur': 1, 'gradient': 1}, {'anyon': 1, 'investig': 1, 'shear': 1, 'buckl': 1, 'stiffen': 1, 'plate': 1}, {'paper': 1, 'shear': 2, 'buckl': 1, 'unstiffen': 1, 'rectangular': 1, 'plate': 1}, {'practic': 1, 'close': 1, 'realiti': 1, 'assumpt': 1, 'flow': 1, 'hyperson': 1, 'shock': 1, 'tube': 1, 'use': 1, 'nitrogen': 1, 'nonvisc': 1, 'thermodynam': 1, 'equilibrium': 1}, {'design': 1, 'factor': 1, 'use': 1, 'control': 1, 'liftdrag': 1, 'ratio': 1, 'mach': 1, 'number': 1}]

- Using the index table, documents length and queries documents are ranked for each query based on the cosine similarity between document and query(by looping through each word in a query) and results are returned per query in this way **(query_id, document_id)** tuple.
  Ex: First 10 documents in ranking for query 1:
  [(1, 958), (1, 1288), (1, 407), (1, 175), (1, 764), (1, 1269), (1, 117), (1, 1220), (1, 367), (1, 1225)

- From the returned ranked documents for all queries and the relevant documents for each query, average precision and average recall is calculated.

-> **Code run instructions:**
Run the python file using the command **python file_name.py** and provide the paths to cranfieldDocs directory, queries.txt and relevance.txt as input parameters when prompted.
Ex:
> python 02-650208577-Garapati.py
Enter document corpus directory path:
./cranfieldDocs/
Enter queries file path:
./queries.txt
Enter relevance queries file path:
./relevance.txt

-> **Determining precision and recall for the queries:**

- **For top 10 documents in the ranking:**

Precision for query 1 for top 10 documents in the ranking: 0.0
Recall for query 1 for top 10 documents in the ranking: 0.0

Precision for query 2 for top 10 documents in the ranking: 0.2
Recall for query 2 for top 10 documents in the ranking: 0.13333333333333333

Precision for query 3 for top 10 documents in the ranking: 0.2
Recall for query 3 for top 10 documents in the ranking: 0.13333333333333333

Precision for query 4 for top 10 documents in the ranking: 0.1
Recall for query 4 for top 10 documents in the ranking: 0.05555555555555555

Precision for query 5 for top 10 documents in the ranking: 0.2
Recall for query 5 for top 10 documents in the ranking: 0.10526315789473684

Precision for query 6 for top 10 documents in the ranking: 0.4
Recall for query 6 for top 10 documents in the ranking: 0.2222222222222222

Precision for query 7 for top 10 documents in the ranking: 0.6
Recall for query 7 for top 10 documents in the ranking: 0.6666666666666666

Precision for query 8 for top 10 documents in the ranking: 0.1
Recall for query 8 for top 10 documents in the ranking: 0.25

Precision for query 9 for top 10 documents in the ranking: 0.0
Recall for query 9 for top 10 documents in the ranking: 0.0

Precision for query 10 for top 10 documents in the ranking: 0.1
Recall for query 10 for top 10 documents in the ranking: 0.041666666666666664

Average precision over 10 queries for top 10 documents in ranking: 0.19000000000000003
Average recall over 10 queries for top 10 documents in ranking: 0.16080409356725148

- **For top 50 documents in the ranking:**

Precision for query 1 for top 50 documents in the ranking: 0.0
Recall for query 1 for top 50 documents in the ranking: 0.0

Precision for query 2 for top 50 documents in the ranking: 0.1
Recall for query 2 for top 50 documents in the ranking: 0.3333333333333333

Precision for query 3 for top 50 documents in the ranking: 0.06
Recall for query 3 for top 50 documents in the ranking: 0.2

Precision for query 4 for top 50 documents in the ranking: 0.06
Recall for query 4 for top 50 documents in the ranking: 0.16666666666666666

Precision for query 5 for top 50 documents in the ranking: 0.1
Recall for query 5 for top 50 documents in the ranking: 0.2631578947368421

Precision for query 6 for top 50 documents in the ranking: 0.1
Recall for query 6 for top 50 documents in the ranking: 0.2777777777777778

Precision for query 7 for top 50 documents in the ranking: 0.16
Recall for query 7 for top 50 documents in the ranking: 0.8888888888888888

Precision for query 8 for top 50 documents in the ranking: 0.06
Recall for query 8 for top 50 documents in the ranking: 0.75

Precision for query 9 for top 50 documents in the ranking: 0.06
Recall for query 9 for top 50 documents in the ranking: 0.375

Precision for query 10 for top 50 documents in the ranking: 0.06
Recall for query 10 for top 50 documents in the ranking: 0.125

Average precision over 10 queries for top 50 documents in ranking: 0.07600000000000003
Average recall over 10 queries for top 50 documents in ranking: 0.3379824561403509

- **For top 100 documents in the ranking:**

Precision for query 1 for top 100 documents in the ranking: 0.0
Recall for query 1 for top 100 documents in the ranking: 0.0

Precision for query 2 for top 100 documents in the ranking: 0.09
Recall for query 2 for top 100 documents in the ranking: 0.6

Precision for query 3 for top 100 documents in the ranking: 0.09
Recall for query 3 for top 100 documents in the ranking: 0.6

Precision for query 4 for top 100 documents in the ranking: 0.06
Recall for query 4 for top 100 documents in the ranking: 0.3333333333333333

Precision for query 5 for top 100 documents in the ranking: 0.13
Recall for query 5 for top 100 documents in the ranking: 0.6842105263157895

Precision for query 6 for top 100 documents in the ranking: 0.09
Recall for query 6 for top 100 documents in the ranking: 0.5

Precision for query 7 for top 100 documents in the ranking: 0.08
Recall for query 7 for top 100 documents in the ranking: 0.8888888888888888

Precision for query 8 for top 100 documents in the ranking: 0.03
Recall for query 8 for top 100 documents in the ranking: 0.75

Precision for query 9 for top 100 documents in the ranking: 0.05
Recall for query 9 for top 100 documents in the ranking: 0.625

Precision for query 10 for top 100 documents in the ranking: 0.04
Recall for query 10 for top 100 documents in the ranking: 0.16666666666666666

Average precision over 10 queries for top 100 documents in ranking: 0.066
Average recall over 10 queries for top 100 documents in ranking: 0.5148099415204679

- **For top 500 documents in the ranking:**

Precision for query 1 for top 500 documents in the ranking: 0.0
Recall for query 1 for top 500 documents in the ranking: 0.0

Precision for query 2 for top 500 documents in the ranking: 0.03
Recall for query 2 for top 500 documents in the ranking: 1.0

Precision for query 3 for top 500 documents in the ranking: 0.03
Recall for query 3 for top 500 documents in the ranking: 1.0

Precision for query 4 for top 500 documents in the ranking: 0.032
Recall for query 4 for top 500 documents in the ranking: 0.8888888888888888

Precision for query 5 for top 500 documents in the ranking: 0.038
Recall for query 5 for top 500 documents in the ranking: 1.0

Precision for query 6 for top 500 documents in the ranking: 0.036
Recall for query 6 for top 500 documents in the ranking: 1.0

Precision for query 7 for top 500 documents in the ranking: 0.018
Recall for query 7 for top 500 documents in the ranking: 1.0

Precision for query 8 for top 500 documents in the ranking: 0.008
Recall for query 8 for top 500 documents in the ranking: 1.0

Precision for query 9 for top 500 documents in the ranking: 0.016
Recall for query 9 for top 500 documents in the ranking: 1.0

Precision for query 10 for top 500 documents in the ranking: 0.024
Recall for query 10 for top 500 documents in the ranking: 0.5

Average precision over 10 queries for top 500 documents in ranking: 0.023200000000000002
Average recall over 10 queries for top 500 documents in ranking: 0.8388888888888889