

1: The steps I have performed for data preprocessing.

I read the smg files parallelly using threads. Each thread read the file article by article. Since each article is surrounded with <REUTERS> tag, I used the substring and find() methods of strings. After storing the article in a variable, I extracted the “NEWID”, <TITLE> and <BODY> parts using the same methodology. If none were found, I assumed it was an empty string.

After merging the title and the body, I did lowercase every character with “lower()” method, removed “\n”s, removed stopwords with “replace()” method, and finally removed punctuations using “translate()” method and “string.punctuation” list.

Then, I started indexing. I started little by little. First, I inverted articles. While inverting the articles, I did not take frequency into account. Final form of the inverted article looks like this: {"term_1": ID, "term_2": ID, "term_3": ID, "term_4": ID,...}.

Secondly, I merged articles of a file. Final form of the merged articles is like this:

{"term_1": [ID1, ID2,..., ID989], "term_2": [ID1, ID54], "term_3": [ID1, ID2, ..., ID45,..., ID876], ...}. Finally, I merged these files into one by simply appending starting from the first file to the last in ascending order. As a result, the final index became automatically sorted without any extra sorting mechanism.

Finally, I dumped this index to a json file named “*myindex_unique.json*”. I did neither indent nor sorted it since it was meant to be read by another python program not a human.

2: The data structures that I used for representing the inverted index

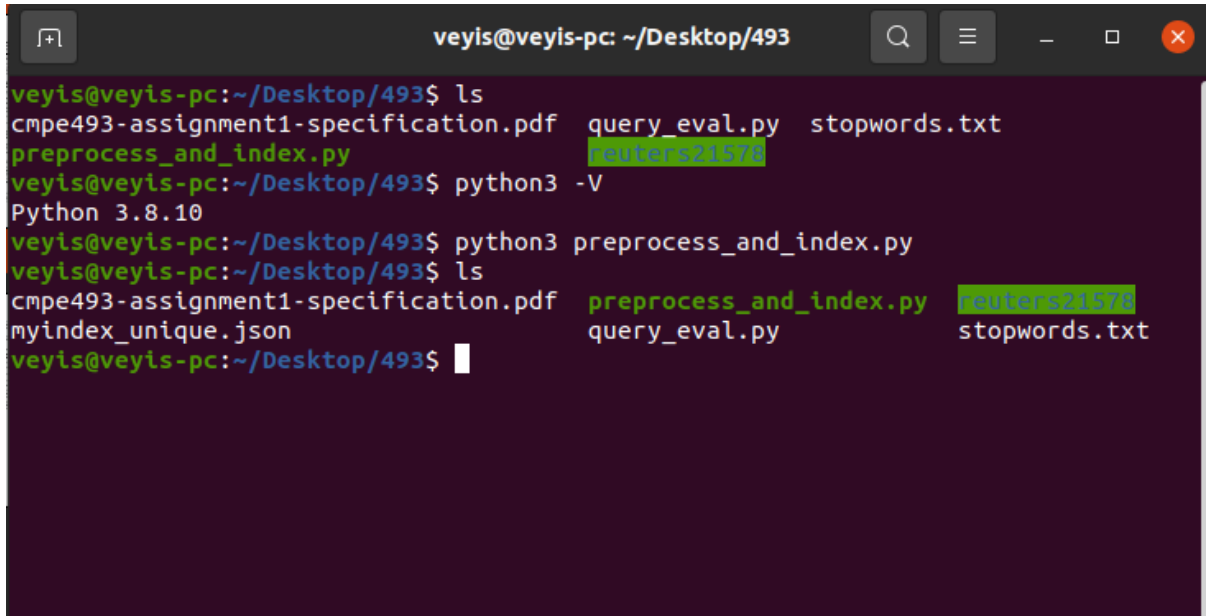
I used python’s built-in “list” collection for posting lists. Appending one list to another and traversing both lists while merging or intersecting two lists into one were simple. I kept it sorted all the time.

I used python’s built-in “dictionary” collection for representing the inverted index. Key values are terms --lowercased and stopwords removed-- and values are the corresponding posting lists.

I believe this approach can be extended with new articles easily. We will just append the new ID’s to the end of the current posting lists. But if we were to change an existing article, it would be costly.

The index is constructed approximately within 6 seconds on my computer.

3: Provide a screenshot of running the indexing module of your system.

A screenshot of a terminal window titled 'veyis@veyis-pc: ~/Desktop/493'. The terminal shows the following commands and output:

```
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  query_eval.py  stopwords.txt
preprocess_and_index.py                reuters21578
veyis@veyis-pc:~/Desktop/493$ python3 -V
Python 3.8.10
veyis@veyis-pc:~/Desktop/493$ python3 preprocess_and_index.py
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  preprocess_and_index.py  reuters21578
myindex_unique.json                  query_eval.py           stopwords.txt
veyis@veyis-pc:~/Desktop/493$
```

4: Provide four screenshots of running your system for each of the four types of queries.

a: conjunction

```
veyis@veyis-pc: ~/Desktop/493
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  preprocess_and_index.py  Reuters21576
myindex_unique.json                  query_eval.py            stopwords.txt
veyis@veyis-pc:~/Desktop/493$ python3 query_eval.py
What is your search query? (Press q for quit.) price and oil
Result is saved to result_2021-11-21 15:20:53.264282.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) q
Now you can see all the contents.
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  query_eval.py            stopwords.txt
myindex_unique.json                  'result_2021-11-21 15:20:53.264282.json'
preprocess_and_index.py              Reuters21576
veyis@veyis-pc:~/Desktop/493$ cat 'result_2021-11-21 15:20:53.264282.json'
[{"price and oil": [127, 144, 191, 194, 213, 236, 246, 263, 357, 471, 489, 502, 543, 597, 829, 834, 843, 873, 885, 952, 1026, 1349, 1370, 1387, 1711, 1875, 1909, 1990, 2045, 2061, 2068, 2074, 2121, 2132, 2228, 2251, 2383, 2696, 2775, 2828, 2833, 2975, 2998, 3024, 3065, 3174, 3181, 3189, 3249, 3303, 3342, 3389, 3430, 3452, 3455, 3490, 3535, 3563, 3571, 3593, 3798, 3869, 3985, 4005, 4017, 4061, 4174, 4214, 4232, 4453, 4474, 4481, 4546, 4564, 4576, 4584, 4634, 4662, 4679, 4713, 4744, 4835, 4878, 5037, 5061, 5145, 5167, 5179, 5184, 5244, 5255, 5268, 5270, 5273, 5318, 5323, 5389, 5559, 5631, 5761, 5769, 5787, 5851, 5936, 6023, 6086, 6121, 6177, 6201, 6208, 6413, 6656, 6876, 6954, 6994, 6996, 7174, 7200, 7408, 7639, 7643, 7731, 7937, 8015, 8041, 8095, 8131, 8134, 8173, 8209, 8210, 8478, 8606, 8615, 8630, 8820, 8960, 8964, 9031, 9077, 9149, 9156, 9213, 9392, 9462, 9485, 9639, 9650, 9691, 9706, 9733, 9761, 9763, 9799, 9853, 9947, 10078, 10080, 10091, 10168, 10190, 10192, 10228, 10261, 10291, 10306, 10330, 10348, 10385, 10567, 10605, 10649, 10693, 10703, 10845, 10873, 10975, 11083, 11118, 11172, 11177, 11213, 11224, 11232, 11236, 11241, 11273, 11350, 11455, 11711, 11723, 11753, 11768, 11778, 11880, 11882, 11949, 12013, 12050, 12111, 12277, 12279, 12281, 12608, 12647, 12670, 12680, 12791, 12799, 13115, 13236, 13265, 13276, 13281, 13290, 13653, 14183, 14558, 14649, 14708, 14724, 14749, 14833, 14873, 14942, 15038, 15084, 15203, 15212, 15322, 15386, 15389, 15575, 15607, 15635, 15829, 15875, 15939, 16116, 16130, 16195, 16215, 16268, 16483, 16589, 16607, 16649, 16939, 16956, 16991, 17015, 17018, 17101, 17102, 17131, 17161, 17173, 17177, 17254, 17289, 17291, 17294, 17329, 17359, 17385, 17405, 17408, 17409, 17416, 17419, 17429, 17446, 17478, 17519, 17759, 17812, 17816, 17892, 17913, 17929, 17963, 18085, 18193, 18280, 18367, 18403, 18422, 18432, 18448, 18621, 18689, 18738, 18744, 18746, 18754, 18765, 18773, 18776, 18795, 18810, 18840, 19051, 19059, 19069, 19083, 19128, 19193, 19285, 19291, 19397, 19490, 19497, 19499, 19509, 19559, 19588, 19662, 19832, 19927, 19998, 20030, 20095, 20352, 20420, 20666, 20709, 20721, 20919, 20936, 21067, 21076, 21131, 21486]]}]
veyis@veyis-pc:~/Desktop/493$
```

b: disjunction

```
veyis@veyis-pc: ~/Desktop/493
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  query_eval.py            stopwords.txt
myindex_unique.json                  'result_2021-11-21 15:20:53.264282.json'
preprocess_and_index.py              Reuters21576
veyis@veyis-pc:~/Desktop/493$ python3 query_eval.py
What is your search query? (Press q for quit.) warwick or nugget or eradication
Result is saved to result_2021-11-21 15:21:28.744046.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) q
Now you can see all the contents.
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  query_eval.py            stopwords.txt
myindex_unique.json                  'result_2021-11-21 15:20:53.264282.json'
preprocess_and_index.py              'result_2021-11-21 15:21:28.744046.json'
veyis@veyis-pc:~/Desktop/493$ cat 'result_2021-11-21 15:21:28.744046.json'
[{"warwick or nugget or eradication": [443, 447, 449, 798, 2226, 4893, 4943, 5422, 9567, 9591, 9630, 14099, 17224, 17285, 18489, 19917, 20620]]}]
veyis@veyis-pc:~/Desktop/493$
```

c: disjunction and negation

```
veyis@veyis-pc: ~/Desktop/493
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  query_eval.py  Reuters21578
myindex_unique.json                   'result_2021-11-21 15:20:53.264282.json'  stopwords.txt
preprocess_and_index.py                'result_2021-11-21 15:21:28.744046.json'
veyis@veyis-pc:~/Desktop/493$ python3 query_eval.py
What is your search query? (Press q for quit.) price AND oil NOT vegetable
Result is saved to result_2021-11-21 15:24:01.480851.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) q
Now you can see all the contents.
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  'result_2021-11-21 15:21:28.744046.json'
myindex_unique.json                   'result_2021-11-21 15:24:01.480851.json'
preprocess_and_index.py               Reuters21578
query_eval.py                         stopwords.txt
'result_2021-11-21 15:20:53.264282.json'
veyis@veyis-pc:~/Desktop/493$ cat 'result_2021-11-21 15:24:01.480851.json'
[{"price and oil not vegetable": [127, 144, 191, 194, 236, 246, 263, 357, 471, 489, 502, 543, 597, 829, 834, 843, 873, 885, 952, 1026, 1349, 1370, 1387, 1711, 1875, 1909, 1990, 2045, 2061, 2068, 2074, 2121, 2132, 2228, 2251, 2383, 2696, 2775, 2828, 2833, 2975, 2998, 3024, 3065, 3174, 3181, 3189, 3249, 3303, 3342, 3389, 3430, 3452, 3455, 3490, 3535, 3563, 3571, 3593, 3798, 3869, 3985, 4005, 4017, 4061, 4174, 4214, 4232, 4453, 4474, 4481, 4546, 4564, 4576, 4584, 4634, 4662, 4679, 4713, 4744, 4835, 4878, 5037, 5061, 5145, 5167, 5179, 5184, 5244, 5255, 5268, 5270, 5273, 5318, 5323, 5389, 5559, 5631, 5761, 5769, 5787, 5851, 5936, 6023, 6086, 6121, 6177, 6201, 6208, 6413, 6656, 6876, 6954, 6994, 6996, 7174, 7200, 7408, 7639, 7643, 7731, 7937, 8015, 8041, 8095, 8131, 8134, 8173, 8209, 8210, 8478, 8606, 8615, 8630, 8820, 8960, 8964, 9031, 9077, 9149, 9156, 9213, 9392, 9462, 9485, 9639, 9650, 9691, 9706, 9733, 9761, 9763, 9799, 9853, 9947, 10078, 10080, 10091, 10168, 10190, 10192, 10228, 10291, 10306, 10385, 10567, 10605, 10649, 10693, 10703, 10845, 10873, 10975, 11083, 11118, 11172, 11177, 11213, 11224, 11232, 11236, 11241, 11273, 11350, 11455, 11711, 11723, 11753, 11768, 11778, 11880, 11882, 11949, 12013, 12050, 12111, 12277, 12279, 12281, 12608, 12647, 12670, 12680, 12791, 12799, 13115, 13236, 13265, 13276, 13281, 13290, 13653, 14183, 14558, 14649, 14708, 14724, 14749, 14833, 14873, 14942, 15038, 15084, 15203, 15212, 15322, 15386, 15389, 15575, 15607, 15635, 15829, 15875, 15939, 16116, 16130, 16195, 16215, 16268, 16483, 16589, 16607, 16649, 16939, 16956, 16991, 17015, 17018, 17101, 17102, 17131, 17161, 17173, 17177, 17254, 17289, 17291, 17294, 17329, 17359, 17385, 17405, 17408, 17409, 17416, 17419, 17429, 17446, 17478, 17519, 17812, 17816, 17892, 17913, 17929, 17963, 18085, 18193, 18280, 18367, 18422, 18432, 18448, 18621, 18689, 18738, 18744, 18746, 18754, 18765, 18773, 18776, 18795, 18810, 18840, 19051, 19059, 19069, 19083, 19128, 19193, 19285, 19291, 19397, 19490, 19497, 19499, 19509, 19559, 19588, 19662, 19832, 19927, 19998, 20030, 20095, 20352, 20420, 20666, 20709, 20721, 20919, 20936, 21067, 21076, 21131, 21486]]]
veyis@veyis-pc:~/Desktop/493$
```

d: conjunction and negation

```
veyis@veyis-pc: ~/Desktop/493
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf      'result_2021-11-21 15:21:28.744046.json'
myindex_unique.json                       'result_2021-11-21 15:24:01.480851.json'
preprocess_and_index.py                   leuters21578
query_eval.py                             stopwords.txt
'result_2021-11-21 15:20:53.264282.json'
veyis@veyis-pc:~/Desktop/493$ python3 query_eval.py
What is your search query? (Press q for quit.) ore or purchases or inventory not diamond not gold
Result is saved to result_2021-11-21 15:25:52.853193.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) q
Now you can see all the contents.
veyis@veyis-pc:~/Desktop/493$ cat 'result_2021-11-21 15:25:52.853193.json'
[{"ore or purchases or inventory not diamond not gold": [19, 208, 223, 264, 297, 344, 437, 454, 496, 588, 629, 634, 674, 752, 881, 888, 957, 983, 1008, 1140, 1215, 1227, 1235, 1257, 1362, 1385, 1399, 1405, 1470, 1472, 1491, 1528, 1535, 1541, 1543, 1598, 1640, 1647, 1650, 1656, 1666, 1682, 1776, 1777, 1781, 1786, 1822, 1831, 1834, 1872, 1934, 1940, 2156, 2206, 2221, 2352, 2372, 2382, 2434, 2456, 2500, 2526, 2565, 2592, 2618, 2662, 2741, 2775, 2802, 2936, 2942, 2943, 2975, 3010, 3026, 3030, 3190, 3199, 3206, 3225, 3272, 3290, 3323, 3324, 3330, 3340, 3360, 3361, 3375, 3429, 3447, 3453, 3458, 3530, 3571, 3622, 3655, 3711, 3758, 3791, 3902, 3904, 3931, 3949, 4146, 4186, 4213, 4297, 4310, 4354, 4401, 4419, 4478, 4502, 4518, 4541, 4579, 4650, 4659, 4682, 4696, 4728, 4735, 4790, 4811, 4825, 4864, 4884, 4967, 5040, 5125, 5127, 5139, 5148, 5180, 5278, 5342, 5382, 5383, 5400, 5431, 5525, 5640, 5648, 5702, 5735, 5750, 5778, 5788, 5827, 5858, 5941, 6038, 6096, 6103, 6144, 6197, 6223, 6247, 6267, 6289, 6407, 6430, 6432, 6479, 6588, 6653, 6671, 6735, 6787, 6806, 6833, 6843, 6870, 6939, 6989, 7012, 7099, 7200, 7304, 7423, 7440, 7534, 7538, 7600, 7672, 7934, 7941, 8020, 8044, 8060, 8210, 8272, 8283, 8357, 8430, 8432, 8451, 8567, 8621, 8628, 8630, 8664, 8666, 8689, 8706, 8735, 8771, 8850, 8862, 8879, 8882, 8922, 8961, 8999, 9022, 9100, 9167, 9176, 9203, 9219, 9290, 9398, 9473, 9519, 9522, 9559, 9582, 9641, 9758, 9953, 9966, 10043, 10082, 10120, 10122, 10169, 10175, 10197, 10259, 10268, 10292, 10309, 10318, 10367, 10375, 10382, 10403, 10406, 10418, 10462, 10472, 10505, 10586, 10623, 10642, 10675, 10677, 10705, 10752, 10761, 10779, 10809, 10829, 10916, 10925, 10959, 10998, 11076, 11083, 11179, 11224, 11227, 11228, 11262, 11272, 11320, 11445, 11491, 11575, 11639, 11654, 11655, 11693, 11798, 11811, 11813, 11843, 11973, 12002, 12055, 12077, 12136, 12164, 12261, 12270, 12305, 12319, 12325, 12332, 12343, 12373, 12417, 12450, 12453, 12494, 12528, 12603, 12641, 12651, 12655, 12695, 12727, 12778, 12883, 13053, 13099, 13129, 13165, 13208, 13271, 13390, 13411, 13462, 13606, 13670, 13676, 13722, 13738, 13739, 13780, 14070, 14109, 14212, 14302, 14332, 14554, 14594, 14619, 14623, 14813, 14833, 14919, 15008, 15111, 15121, 15263, 15303, 15310, 15386, 15389, 15450, 15724, 15815, 15871, 15893, 15894, 15916, 15960, 15961, 15975, 16010, 16016, 16021, 16026, 16028, 16075, 16086, 16094, 16122, 16177, 16205, 16316, 16320, 16357, 16547, 16659, 16669, 16765, 16774, 16784, 16787, 16844, 16869, 16976, 17003, 17004, 17023, 17037, 17075, 17105, 17107, 17109, 17153, 17154, 17167, 17177, 17179, 17194, 17225, 17248, 17258, 17304, 17306, 17312, 17341, 17387, 17408, 17416, 17480, 17486, 17539, 17546, 17573, 17576, 17707, 17731, 17783, 17805, 17930, 17950, 18014, 18146, 18256, 18276, 18310, 18337, 18343, 18358, 18385, 18397, 18405, 18412, 18413, 18417, 18431, 18462, 18532, 18542, 18557, 18571, 18650, 18676, 18726, 18813, 18817, 18849, 18917, 18920, 18930, 18987, 19040, 19050, 19444, 19477, 19478, 19596, 19683, 19690, 19713, 19715, 19722, 19728, 19731, 19751, 19869, 19873, 20137, 20188, 20199, 20232, 20264, 20405, 20457, 20525, 20585, 20602, 20649, 20665, 20694, 20698, 20766, 20772, 20797, 20815, 20820, 20841, 20873, 20877, 20886, 20910, 20917, 20945, 20967, 21027, 21070, 21089, 21337, 21486]]}]
veyis@veyis-pc:~/Desktop/493$
```


e: mixed

```
veyis@veyis-pc: ~/Desktop/493
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  'result_2021-11-21 15:20:53.264282.json'  Reuters21574
myindex_unique.json                  'result_2021-11-21 15:21:28.744046.json'  stopwords.txt
preprocess_and_index.py              'result_2021-11-21 15:24:01.480851.json'
query_eval.py                       'result_2021-11-21 15:25:52.853193.json'
veyis@veyis-pc:~/Desktop/493$ python3 query_eval.py
What is your search query? (Press q for quit.) milk
Result is saved to result_2021-11-21 15:28:01.864510.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) reform and effects
Result is saved to result_2021-11-21 15:28:01.864510.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) improved not reform not democracy
Result is saved to result_2021-11-21 15:28:01.864510.json file. You can see it after closing the program.
What is your search query? (Press q for quit.) q
Now you can see all the contents.
veyis@veyis-pc:~/Desktop/493$ ls
cmpe493-assignment1-specification.pdf  'result_2021-11-21 15:24:01.480851.json'
myindex_unique.json                  'result_2021-11-21 15:25:52.853193.json'
preprocess_and_index.py              'result_2021-11-21 15:28:01.864510.json'
query_eval.py                       Reuters21574
'result_2021-11-21 15:20:53.264282.json'  stopwords.txt
'result_2021-11-21 15:21:28.744046.json'
veyis@veyis-pc:~/Desktop/493$ cat 'result_2021-11-21 15:28:01.864510.json'
[{"milk": [18, 672, 862, 876, 1439, 1597, 1776, 1813, 2030, 2068, 2077, 2389, 3330, 5761, 5887, 6040, 6444, 8743, 9567, 10124, 10334, 10830, 11544, 12584, 12907, 13452, 14656, 14670, 16093, 16772, 18512, 18866, 18980, 19275, 21152]}, {"reform and effects": [18, 2068, 3502, 3670, 9055, 10049, 10605, 12025, 14749, 16171, 16196, 17242]}, {"improved not reform not democracy": [4, 16, 23, 39, 252, 317, 348, 356, 357, 358, 362, 364, 552, 748, 843, 921, 938, 971, 988, 1019, 1132, 1140, 1193, 1230, 1252, 1269, 1272, 1361, 1466, 1535, 1576, 1685, 1867, 1919, 2116, 2203, 2371, 2435, 2475, 2511, 2569, 2618, 2729, 2764, 2822, 2833, 2879, 3267, 3271, 3349, 3432, 3488, 3490, 3564, 3657, 3841, 4016, 4223, 4232, 4429, 4522, 4608, 4626, 4663, 4692, 4695, 4742, 4795, 5027, 5039, 5071, 5102, 5184, 5210, 5214, 5385, 5431, 5474, 5482, 5500, 5526, 5564, 5599, 5705, 5745, 5787, 5788, 6001, 6153, 6158, 6214, 6406, 6434, 6531, 6722, 6723, 6811, 6903, 6949, 7102, 7174, 7405, 7510, 7606, 7787, 7870, 7871, 7879, 7902, 7904, 7917, 7937, 8003, 8050, 8051, 8101, 8162, 8183, 8271, 8277, 8308, 8328, 8365, 8439, 8441, 8585, 8627, 8799, 8863, 9142, 9192, 9451, 9485, 9641, 9665, 9734, 9782, 9803, 9848, 9874, 9940, 10055, 10082, 10250, 10272, 10494, 10643, 10740, 10841, 10944, 11091, 11144, 11270, 11327, 11436, 11624, 11773, 11776, 11839, 11852, 12044, 12065, 12070, 12092, 12420, 12479, 12496, 12573, 12637, 12744, 12799, 12833, 12986, 13115, 13174, 13178, 13258, 13313, 13321, 13379, 13491, 13543, 13721, 14212, 14279, 14310, 14314, 14342, 14354, 14390, 14691, 14719, 14828, 14832, 14869, 14916, 14942, 14946, 14993, 15093, 15162, 15286, 15345, 15389, 15396, 15466, 15567, 15574, 15621, 15639, 15656, 15840, 16028, 16126, 16140, 16220, 16227, 16252, 16336, 16577, 16594, 16775, 16824, 16826, 16853, 16895, 16998, 17097, 17136, 17327, 17352, 17369, 17384, 17405, 17460, 17541, 17622, 17632, 17708, 17721, 17865, 17923, 17937, 18143, 18147, 18240, 18252, 18380, 18409, 18455, 18532, 18849, 19030, 19037, 19038, 19075, 19259, 19332, 19478, 19507, 19707, 19724, 19779, 19797, 19814, 19958, 20029, 20272, 20295, 20511, 20530, 20782, 20906, 21029, 21145, 21147, 21223, 21500, 21534]]}]
veyis@veyis-pc:~/Desktop/493$
```