

```
In [16]: import numpy as np
print(a)
[[ 1.  2.  3.  4.]
 [ 5.  6.  7.  8.]
 [ 9. 10. 11. 12.]
 [13. 14. 15. 16]]

In [17]: b=a
print(b)
[[ 1.  2.  3.  4.]
 [ 5.  6.  7.  8.]
 [ 9. 10. 11. 12.]
 [13. 14. 15. 16]]

In [18]: print(a)
[[ 1.  2.  3.  4.]
 [ 5.  6.  7.  8.]
 [ 9. 10. 11. 12.]
 [13. 14. 15. 16]]

In [19]: c=np.array([[4,3,2,1],[6,7,6,5],[12,11,16,9],[16,15,14,13]])
print(c)
[[ 4.  3.  2.  1.]
 [ 6.  7.  6.  5.]
 [12. 11. 16.  9.]
 [16. 15. 14. 13]]

In [ ]:

In [20]: c.T
Out[20]: array([[ 4,  8, 12, 16],
               [ 3,  7, 11, 15],
               [ 2,  6, 10, 14],
               [ 1,  5,  9, 13]])

In [22]: c[:,1]
Out[22]: array([ 3,  7, 11, 15])

In [23]: b=c
Out[23]: array([[ 5,  5,  5,  5],
               [13, 13, 13, 13],
               [21, 21, 21, 21],
               [29, 29, 29, 29]])

In [24]: np.add(b,c)
Out[24]: array([[ 9,  5,  5,  5],
               [13, 13, 13, 13],
               [21, 21, 21, 21],
               [29, 29, 29, 29]])

In [25]: np.subtract(b,c)
Out[25]: array([[ -3, -1, -1,  3],
               [-3, -1,  3,  3],
               [-3, -1,  3,  3],
               [-3, -1, -1, -3]])

In [26]: b=c
Out[26]: array([[ -3, -1, -1,  3],
               [-3, -1, -1,  3],
               [-3, -1,  3,  3],
               [-3, -1,  3,  3]])

In [27]: b=c
Out[27]: array([[ 4,  6,  6,  4],
               [40, 42, 42, 40],
               [108, 110, 110, 108],
               [208, 210, 210, 208]])

In [29]: np.multiply(b,c)
Out[29]: array([[ 4,  6,  6,  4],
               [40, 42, 42, 40],
               [108, 110, 110, 108],
               [208, 210, 210, 208]])

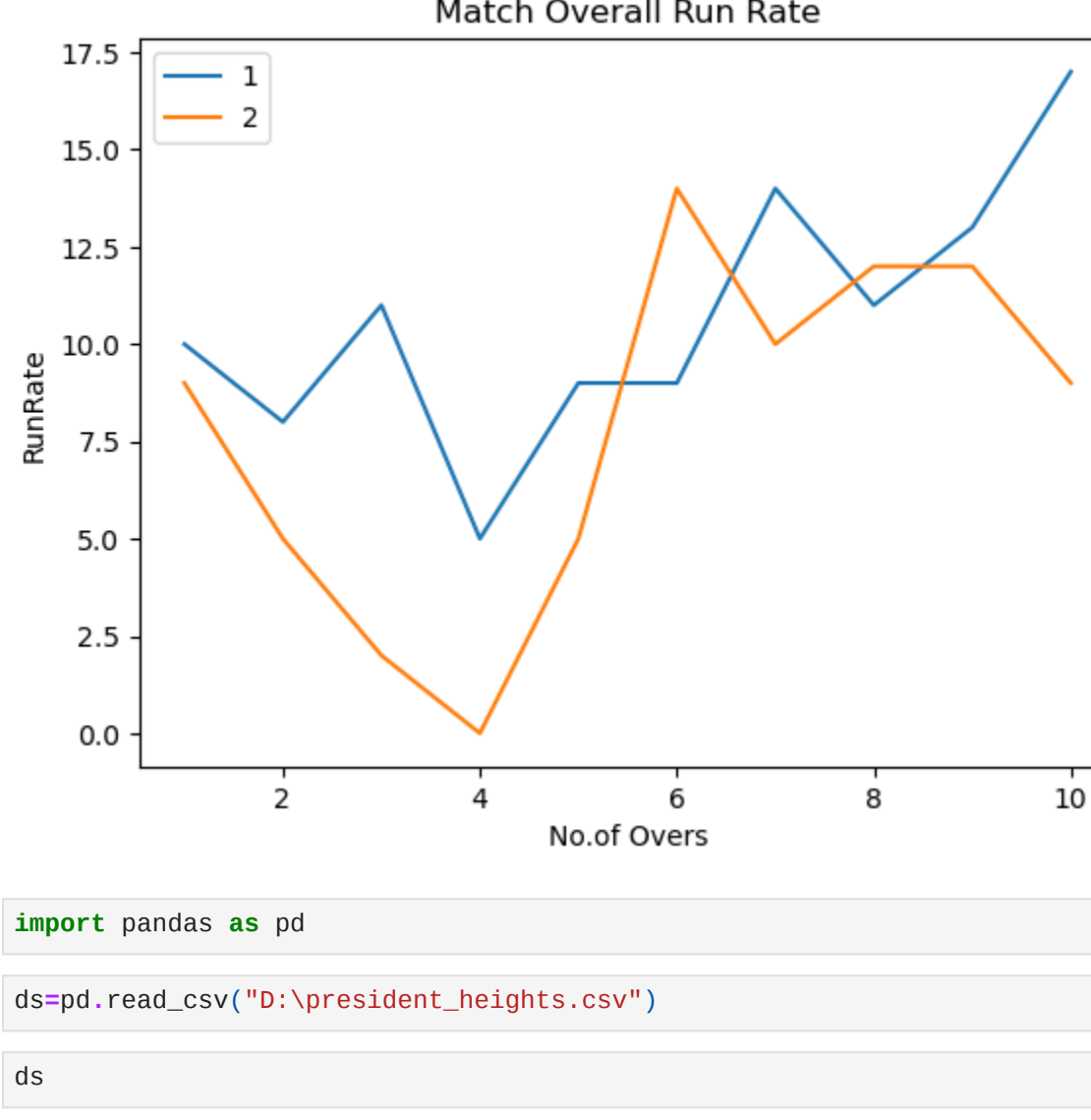
In [30]: np.divide(b,c)
Out[30]: array([[0.25,  0.66666667, 1.5,  4.],
               [0.625,  0.85714286, 1.66666667, 1.6],
               [0.75,  0.90909091, 1.1,  1.33333333],
               [0.8125,  0.83333333, 1.07142857, 1.28571429]])

In [31]: b/c
Out[31]: array([[0.25,  0.66666667, 1.5,  4.],
               [0.625,  0.85714286, 1.66666667, 1.6],
               [0.75,  0.90909091, 1.1,  1.33333333],
               [0.8125,  0.83333333, 1.07142857, 1.28571429]])


In [32]: np.sqrt(b)
Out[32]: array([[ 2.0,  2.44948974, 2.44948974, 2.0],
               [ 2.23606798,  2.44948974, 2.44948974, 2.0],
               [ 3.0,  3.0227766,  3.0227766,  3.0],
               [ 3.89551229,  3.74165738,  3.74165738,  3.89551229]])

In [33]: import matplotlib.pyplot as plt

In [34]: x=np.arange(0,5,0.1)
y=np.sin(x)
plt.plot(x,y)
plt.xlabel("RunRate")



In [35]: plt.plot([1,2,3,4,5,6,7,8,9,10],
              [10,12,8,11,5,9,14,11,13,17],
              label='RunRate')
plt.xlabel("No. of Overs")
plt.ylabel("RunRate")
plt.title("Match Overall Run Rate")
plt.show()



In [36]: import pandas as pd

In [37]: ds=pd.read_csv("D:\president_heights.csv")

In [38]: ds
Out[38]:
   order  name  height(cm)
0      1  George Washington    189
1      2    John Adams        170
2      3  Thomas Jefferson    189
3      4    James Madison    163
4      5    James Monroe    183
5      6  John Quincy Adams    171
6      7    Andrew Jackson    185
7      8  Martin Van Buren    168
8      9  William Henry Harrison    173
9     10    John Tyler        183
10     11  James K. Polk        173
11     12  Zachary Taylor        173
12     13  Millard Fillmore    175
13     14  Franklin Pierce        178
14     15  James Buchanan    183
15     16  Abraham Lincoln    193
16     17    Andrew Johnson    178
17     18  Ulysses S. Grant        173
18     19  Rutherford B. Hayes    174
19     20  James A. Garfield    183
20     21  Chester A. Arthur    183
21     22  Benjamin Harrison    168
22     23  William McKinley    170
23     24  Theodore Roosevelt    178
24     25  William Howard Taft    182
25     26  Woodrow Wilson        180
26     27  Warren G. Harding    183
27     28  Calvin Coolidge        178
28     29  Herbert Hoover        182
29     30  Franklin D. Roosevelt    188
30     31  Harry S. Truman        175
31     32  Dwight D. Eisenhower    179
32     33  John F. Kennedy        183
33     34  Lyndon B. Johnson    193
34     35  Richard Nixon        182
35     36  Gerald Ford        183
36     37  Jimmy Carter        177
37     38  Ronald Reagan        185
38     39  George H. W. Bush    188
39     40  Bill Clinton        188
40     41  George W. Bush        182
41     42  Barack Obama        185

In [39]: ds.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42 entries, 0 to 41
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0   order  42 non-null      int64
 1   name    42 non-null      object
 2   height(cm)  42 non-null     object
dtypes: int64(1), object(2)
memory usage: 1.1+ KB

In [40]: ds.dtypes
Out[40]:
order      int64
name       object
height(cm) int64
dtype: object

In [41]: ds1=ds
Out[41]:
   order  name  height(cm)
0      1  George Washington    189
1      2    John Adams        170
2      3  Thomas Jefferson    189
3      4    James Madison    163
4      5    James Monroe    183
5      6  John Quincy Adams    171
6      7    Andrew Jackson    185
7      8  Martin Van Buren    168
8      9  William Henry Harrison    173
9     10    John Tyler        183
10     11  James K. Polk        173
11     12  Zachary Taylor        173
12     13  Millard Fillmore    175
13     14  Franklin Pierce        178
14     15  James Buchanan    183
15     16  Abraham Lincoln    193
16     17    Andrew Johnson    178
17     18  Ulysses S. Grant        173
18     19  Rutherford B. Hayes    174
19     20  James A. Garfield    183
20     21  Chester A. Arthur    183
21     22  Benjamin Harrison    168
22     23  William McKinley    170
23     24  Theodore Roosevelt    178
24     25  William Howard Taft    182
25     26  Woodrow Wilson        180
26     27  Warren G. Harding    183
27     28  Calvin Coolidge        178
28     29  Herbert Hoover        182
29     30  Franklin D. Roosevelt    188
30     31  Harry S. Truman        175
31     32  Dwight D. Eisenhower    179
32     33  John F. Kennedy        183
33     34  Lyndon B. Johnson    193
34     35  Richard Nixon        182
35     36  Gerald Ford        183
36     37  Jimmy Carter        177
37     38  Ronald Reagan        185
38     39  George H. W. Bush    188
39     40  Bill Clinton        188
40     41  George W. Bush        182
41     42  Barack Obama        185

In [42]: ds1.head(10)
Out[42]:
   order  name  height(cm)
0      1  George Washington    189
1      2    John Adams        170
2      3  Thomas Jefferson    189
3      4    James Madison    163
4      5    James Monroe    183
5      6  John Quincy Adams    171
6      7    Andrew Jackson    185
7      8  Martin Van Buren    168
8      9  William Henry Harrison    173
9     10    John Tyler        183

In [43]: ds.tail(10)
Out[43]:
   order  name  height(cm)
37     38  Ronald Reagan        185
38     41  George H. W. Bush    188
39     42  Bill Clinton        188
40     43  George W. Bush        182
41     44  Barack Obama        185

In [44]: ds1.head(1)
Out[44]:
   order  name  height(cm)
0      1  George Washington    189

In [45]: ds1.loc[35]
Out[45]:
order      35
name       Gerald Ford
height(cm) 183
Name: 35, dtype: object

In [46]: ds1.loc[35:39]
Out[46]:
   order  name  height(cm)
36     36  Gerald Ford        183
37     37  Jimmy Carter        177
38     38  Ronald Reagan        185
39     41  George H. W. Bush    188

In [47]: ds1.describe()
Out[47]:
   order  height(cm)
count    42.000000    42.000000
mean     22.478190    179.730955
std       11.550451     7.050909
min        1.000000    163.000000
25%       11.250000    174.250000
50%       22.000000    182.000000
75%       33.750000    183.000000
max       44.000000    193.000000

In [48]: ds1[order].describe()
Out[48]:
count    42.000000
mean     22.478190
std       11.550451
min        1.000000
25%       11.250000
50%       22.000000
75%       33.750000
max       44.000000
Name: order, dtype: float64

In [49]: subset=ds1[['order','name']]
Out[49]:
   order  name
0      1  George Washington
1      2    John Adams
2      3  Thomas Jefferson
3      4    James Madison
4      5    James Monroe
5      6  John Quincy Adams
6      7    Andrew Jackson
7      8  Martin Van Buren
8      9  William Henry Harrison
9     10    John Tyler
10     11  James K. Polk
11     12  Zachary Taylor
12     13  Millard Fillmore
13     14  Franklin Pierce
14     15  James Buchanan
15     16  Abraham Lincoln
16     17    Andrew Johnson
17     18  Ulysses S. Grant
18     19  Rutherford B. Hayes
19     20  James A. Garfield
20     21  Chester A. Arthur
21     22  Benjamin Harrison
22     23  William McKinley
23     24  Theodore Roosevelt
24     25  William Howard Taft
25     26  Woodrow Wilson
26     27  Warren G. Harding
27     28  Calvin Coolidge
28     29  Herbert Hoover
29     30  Franklin D. Roosevelt
30     31  Harry S. Truman
31     32  Dwight D. Eisenhower
32     33  John F. Kennedy
33     34  Lyndon B. Johnson
34     35  Richard Nixon
35     36  Gerald Ford
36     37  Jimmy Carter
37     38  Ronald Reagan
38     41  George H. W. Bush
39     42  Bill Clinton
40     43  George W. Bush
41     44  Barack Obama

In [50]: ds1.columns
Out[50]: Index(['order', 'name', 'height(cm)', dtype='object')

In [51]: ds1
Out[51]:
   order  name  height(cm)
0      1  George Washington    189
1      2    John Adams        170
2      3  Thomas Jefferson    189
3      4    James Madison    163
4      5    James Monroe    183
5      6  John Quincy Adams    171
6      7    Andrew Jackson    185
7      8  Martin Van Buren    168
8      9  William Henry Harrison    173
9     10    John Tyler        183
10     11  James K. Polk        173
11     12  Zachary Taylor        173
12     13  Millard Fillmore    175
13     14  Franklin Pierce        178
14     15  James Buchanan    183
15     16  Abraham Lincoln    193
16     17    Andrew Johnson    178
17     18  Ulysses S. Grant        173
18     19  Rutherford B. Hayes    174
19     20  James A. Garfield    183
20     21  Chester A. Arthur    183
21     22  Benjamin Harrison    168
22     23  William McKinley    170
23     24  Theodore Roosevelt    178
24     25  William Howard Taft    182
25     26  Woodrow Wilson        180
26     27  Warren G. Harding    183
27     28  Calvin Coolidge        178
28     29  Herbert Hoover        182
29     30  Franklin D. Roosevelt    188
30     31  Harry S. Truman        175
31     32  Dwight D. Eisenhower    179
32     33  John F. Kennedy        183
33     34  Lyndon B. Johnson    193
34     35  Richard Nixon        182
35     36  Gerald Ford        183
36     37  Jimmy Carter        177
37     38  Ronald Reagan        185
38     41  George H. W. Bush    188
39     42  Bill Clinton        188
40     43  George W. Bush        182
41     44  Barack Obama        185

In [52]: ds1.insert(7,"8","Dr. Rikhanth the Emperor","178")
IndexError: index 7 is out of bounds for axis 0 with size 9
Traceback (most recent call last):
  File "C:\Users\user\AppData\Local\Temp\ipykernel_7468\3388997478.py", line 1, in <module>
    ds1.insert(7,"8","Dr. Rikhanth the Emperor","178")
  File ~\anaconda3\lib\site-packages\pandas\core\frame.py in insert(self, loc, column, value, allow_duplicates)
    4440         value = self._sanitize_column(value)
    4441     self._mgr.insert(loc, column, value)
    4442     def assign(self, **kwargs) -> DataFrame:
    4443         """
    4444         ~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in insert(self, loc, item, value)
    1239         # insert to the axis; this could possibly raise a TypeError
    1240         new_values = self.items.insert(loc, item)
    1241         if value.ndim == 2:
    1242             # special casing: adding 0/1/2/3/4 https://github.com/numpy/numpy/issues/12550
    1243             casted = arr.dtype.type(item)
    1244             new_values = np.insert(arr, loc, casted)
    1245         else:
    1246             # array_function...internals in insert(*args,**kwargs)
    1247         ~\anaconda3\lib\site-packages\numpy.lib.function_base.py in insert(arr, obj, values, axis)
    4783         if index < -N or index > N:
    4784             raise IndexError
    4785         "index %i is out of bounds for axis %i with "
    4786         "size %i" % (obj, axis, N)
    4787
    4788
    4789
    4790
    4791
    4792
    4793
    4794
    4795
    4796
    4797
    4798
    4799
    4800
    4801
    4802
    4803
    4804
    4805
    4806
    4807
    4808
    4809
    4810
    4811
    4812
    4813
    4814
    4815
    4816
    4817
    4818
    4819
    4820
    4821
    4822
    4823
    4824
    4825
    4826
    4827
    4828
    4829
    4830
    4831
    4832
    4833
    4834
    4835
    4836
    4837
    4838
    4839
    4840
    4841
    4842
    4843
    4844
    4845
    4846
    4847
    4848
    4849
    4850
    4851
    4852
    4853
    4854
    4855
    4856
    4857
    4858
    4859
    4860
    4861
    4862
    4863
    4864
    4865
    4866
    4867
    4868
    4869
    4870
    4871
    4872
    4873
    4874
    4875
    4876
    4877
    4878
    4879
    4880
    4881
    4882
    4883
    4884
    4885
    4886
    4887
    4888
    4889
    4890
    4891
    4892
    4893
    4894
    4895
    4896
    4897
    4898
    4899
    4900
    4901
    4902
    4903
    4904
    4905
    4906
    4907
    4908
    4909
    4910
    4911
    4912
    4913
    4914
    4915
    4916
    4917
    4918
    4919
    4920
    4921
    4922
    4923
    4924
    4925
    4926
    4927
    4928
    4929
    4930
    4931
    4932
    4933
    4934
    4935
    4936
    4937
    4938
    4939
    4940
    4941
    4942
    4943
    4944
    4945
    4946
    4947
    4948
    4949
    4950
    4951
    4952
    4953
    4954
    4955
    4956
    4957
    4958
    4959
    4960
    4961
    4962
    4963
    4964
    4965
    4966
    4967
    4968
    4969
    4970
    4971
    4972
    4973
    4974
    4975
    4976
    4977
    4978
    4979
    4980
    4981
    4982
    4983
    4984
    4985
    4986
    4987
    4988
    4989
    4990
    4991
    4992
    4993
    4994
    4995
    4996
    4997
    4998
    4999
    5000
    5001
    5002
    5003
    5004
    5005
    5006
    5007
    5008
    5009
    5010
    5011
    5012
    5013
    5014
    5015
    5016
    5017
    5018
    5019
    5020
    5021
    5022
    5023
    5024
    5025
    5026
    5027
    5028
    5029
    5030
    5031
    5032
    5033
    5034
    5035
    5036
    5037
    5038
    5039
    5040
    5041
    5042
    5043
    5044
    5045
    5046
    5047
    5048
    5049
    5050
    5051
    5052
    5053
    5054
    5055
    5056
    5057
    5058
    5059
    5060
    5061
    5062
    5063
    5064
    5065
    5066
    5067
    5068
    5069
    5070
    5071
    5072
    5073
    5074
    5075
    5076
    5077
    5078
    5079
    5080
    5081
    5082
    5083
    5084
    5085
    5086
    5087
    5088
    5089
    5090
    5091
    5092
    5093
    5094
    5095
    5096
    5097
    5098
    5099
    5100
    5101
    5102
    5103
    5104
    5105
    5106
    5107
    5108
    5109
    5110
    5111
    5112
    5113
    5114
    5115
    5116
    5117
    5118
    5119
    5120
    5121
    5122
    5123
    5124
    5125
    5126
    5127
    5128
    5129
    5130
    5131
    5132
    5133
    5134
    5135
    5136
    5137
    5138
    5139
    5140
    5141
    5142
    5143
    5144
    5145
    5146
    5147
    5148
    5149
    5150
    5151
    5152
    5153
    5154
    5155
    5156
    5157
    5158
    5159
    5160
    5161
    5162
    5163
    5164
    5165
    5166
    5167
    5168
    5169
    5170
    5171
    5172
    5173
    5174
    5175
    5176
    5177
    5178
    5179
    5180
    5181
    5182
    5183
    5184
    5185
    5186
    5187
    5188
    5189
    5190
    5191
    5192
    5193
    5194
    5195
    5196
    5197
    5198
    5199
    5200
    5201
    5202
    5203
    5204
    5205
    5206
    5207
    5208
    5209
    5210
    5211
    5212
    5213
    5214
    5215
    5216
    5217
    5218
    5219
    5220
    5221
    5222
    5223
    5224
    5225
    5226
    5227
    5228
    5229
    5230
    5231
    5232
    5233
    5234
    5235
    5236
    5237
    5238
    5239
    5240
    5241
    5242
    5243
    5244
    5245
    5246
    5247
    5248
    5249
    5250
    5251
    5252
    5253
    5254
    5255
    5256
    5257
    5258
    5259
    5260
    5261
    5262
    5263
    5264
    5265
    5266
    5267
    5268
    5269
    5270
    5271
    5272
    5273
    5274
    5275
    5276
    5277
    5278
    5279
    5280
    5281
    5282
    5283
    5284
    5285
    5286
    5287
    5288
    5289
    5290
    5291
    5292
    5293
    5294
    5295
    5296
    5297
    5298
    5299
    5300
    5301
    5302
    5303
    5304
    5305
    5306
    5307
    5308
    5309
    5310
    5311
    5312
    5313
    5314
    5315
    5316
    5317
    5318
    5319
    5320
    5321
    5322
    5323
    5324
    5325
    5326
    5327
    5328
    5329
    5330
    5331
    5332
    5333
    5334
    5335
    5336
    5337
    5338
    5339
    5340
    5341
    5342
    5343
    5344
    5345
    5346
    5347
    5348
    5349
    5350
    5351
    5352
    5353
    5354
    5355
    5356
    5357
    5358
    5359
    5360
    5361
    5362
    5363
    5364
    5365
    5366
    5367
    5368
    5369
    5370
    5371
    5372
    5373
    5374
    5375
    5376
    5377
    5378
    5379
    5380
    5381
    5382
    5383
    5384
    5385
    5386
    5387
    5388
    5389
    5390
    5391
    5392
    5393
    5394
    5395
    5396
    5397
    5398
    5399
    5400
    5401
    5402
    5403
    5404
    5405
    5406
    5407
    5408
    5409
    5410
    5411
    5412
    5413
    5414
    5415
    5416
    5417
    5418
    5419
    5420
    5421
    5422
    5423
    5424
    5425
    5426
    5427
    5428
    5429
    5430
    5431
    5432
    5433
    5434
    5435
    5436
    5437
    5438
    5439
    5440
    5441
    5442
    5443
    5444
    5445
    5446
    5447
    5448
    5449
    5450
    5451
    5452
    5453
    5454
    5455
    5456
    5457
    5458
    5459
    5460
    5461
    5462
    5463
    5464
    5465
    5466
    5467
    5468
    5469
    5470
    5471
    5472
    5473
    5474
    5475
    5476
    5477
    5478
    5479
    5480
    5481
    5482
    5483
    5484
    5485
    5486
    5487
    5488
    5489
    5490
    5491
    5492
    5493
    5494
    5495
    5496
    5497
    5498
    5499
    5500
    5501
    5502
    5503
    5504
    5505
    5506
    5507
    5508
    5509
    5510
    5511
    5512
    5513
    5514
    5515
    5516
    5517
    5518
    5519
    5520
    5521
    5522
    5523
    5524
    5525
    5526
    5527
    5528
    5529
    5530
    5531
    5532
    5533
    5534
    5535
    5536
    5537
    5538
    5539
    5540
    5541
    5542
    5543
    5544
    5545
    5546
    5547
    5548
    5549
    5550
    5551
    5552
    5553
    5554
    5555
    5556
    5557
    5558
    5559
    5560
    5561
    5562
    5563
    5564
    5565
    5566
    5567
    5568
    5569
    5570
    5571
    5572
    5573
    5574
    5575
    5576
    5577
    5578
    5579
    5580
    5581
    5582
    5583
    5584
    5585
    5586
    5587
    5588
    5589
    5590
    5591
    5592
    5593
    5594
    5595
    5596
    5597
    5598
    5599
    5600
    5601
    5602
    5603
    5604
    5605
    5606
    5607
    5608
    5609
    5610
    5611
    5612
    5613
    5614
    5615
    5616
    5617
    5618
    5619
    5620
    5621
    5622
    5623
    5624
    5625
    5626
    5627
    5628
    5629
    5630
    5631
    5632
    5633
    5634
    5635
    5636
    5637
    5638
    5639
    5640
    5641
    5642
    5643
    5644
    5645
    5646
    5647
    5648
    5649
    5650
    5651
    5652
    5653
    5654
    5655
    5656
    5657
    5658
    5659
    5660
    5661
    5662
    5663
    5664
    5665
    5666
    5667
    5668
    5669
    5670
    5671
    5672
    5673
    5674
    5675
    5676
    5677
    5678
    5679
    5680
    5681
    5682
    5683
    5684
    5685
    5686
    5687
    5688
   
```