

SUMESH R - 20104169

Basic Analysis using NumPy and Pandas

Import Libraries

```
In [1]: import pandas as pd
```

```
In [2]: import numpy as np
```

Import Dataset

```
In [3]: data = pd.read_csv("6_Salesworkload1.csv")
```

```
In [4]: display(data)
```

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	0.0	
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0.0	
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0.0	
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	0.0	
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	0.0	
...
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	0.0	3
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	0.0	
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	0.0	
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	0.0	
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	0.0	3

7658 rows × 14 columns



To display top 10 rows

In [5]:

```
data.head(10)
```

Out[5]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sale unit
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	0.0	398560.0
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0.0	82725.0
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0.0	438400.0
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	0.0	309425.0
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	0.0	165515.0
5	10.2016	1.0	United Kingdom	88253.0	London (I)	6.0	Meat	8270.316	0.0	1713310.0
6	10.2016	1.0	United Kingdom	88253.0	London (I)	13.0	Food	16468.251	0.0	3107935.0
7	10.2016	1.0	United Kingdom	88253.0	London (I)	7.0	Clothing	4698.471	0.0	213680.0
8	10.2016	1.0	United Kingdom	88253.0	London (I)	8.0	Household	1183.272	0.0	54915.0
9	10.2016	1.0	United Kingdom	88253.0	London (I)	9.0	Hardware	2029.815	0.0	59260.0

to display last 5 rows

In [6]:

```
data.tail()
```

Out[6]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	0.0
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	0.0
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	0.0
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	0.0
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	0.0

statistical summary

In [7]:

```
data.describe()
```

Out[7]:

	Time index	StoreID	Dept_ID	HoursLease	Sales units	Turnover	Customer
count	7650.000000	7650.000000	7650.000000	7650.000000	7.650000e+03	7.650000e+03	0.0
mean	5.000000	61995.220000	9.470588	22.036078	1.076471e+06	3.721393e+06	NaN
std	2.582158	29924.581631	5.337429	133.299513	1.728113e+06	6.003380e+06	NaN
min	1.000000	12227.000000	1.000000	0.000000	0.000000e+00	0.000000e+00	NaN
25%	3.000000	29650.000000	5.000000	0.000000	5.457125e+04	2.726798e+05	NaN
50%	5.000000	75400.500000	9.000000	0.000000	2.932300e+05	9.319575e+05	NaN
75%	7.000000	87703.000000	14.000000	0.000000	9.175075e+05	3.264432e+06	NaN
max	9.000000	98422.000000	18.000000	3984.000000	1.124296e+07	4.271739e+07	NaN

To print number of elements

In [8]:

```
data.size
```

Out[8]: 107212

to print number of row and cols

In [9]:

```
data.shape
```

Out[9]: (7658, 14)

to find missing values

In [10]:

```
data.isna()
```

Out[10]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sales units	Turn
0	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	F
...

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sales units	Turnc
7653	False	False	False	False	False	False	False	False	False	False	F
7654	False	False	False	False	False	False	False	False	False	False	F
7655	False	False	False	False	False	False	False	False	False	False	F
7656	False	False	False	False	False	False	False	False	False	False	F
7657	False	False	False	False	False	False	False	False	False	False	F

7658 rows × 14 columns

fill null values with a constant

In [11]:

```
data.fillna(5)
```

Out[11]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease		
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	0.0		
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0.0		
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0.0		
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	0.0		
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	0.0		
...
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	0.0	3	
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	0.0		
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	0.0		
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	0.0		
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	0.0	3	

7658 rows × 14 columns

to select a particular columns

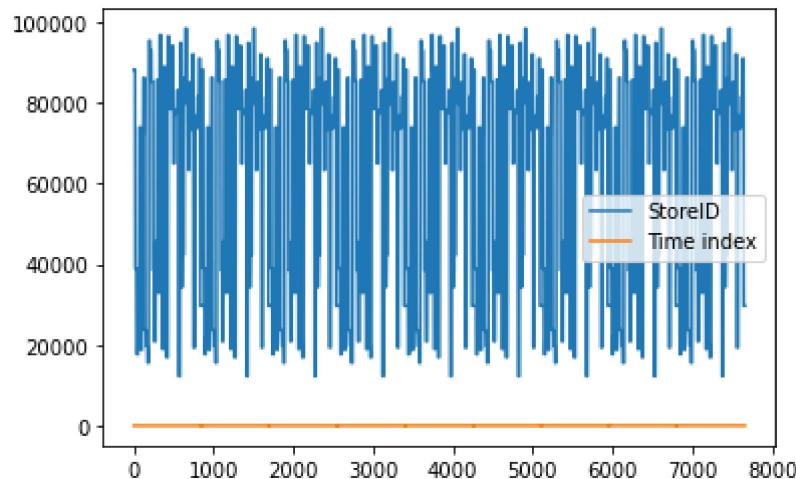
In [12]:

```
df=pd.DataFrame(data[['StoreID','Time index']])
import matplotlib.pyplot as plt
```

line plot

In [13]: `df.plot.line()`

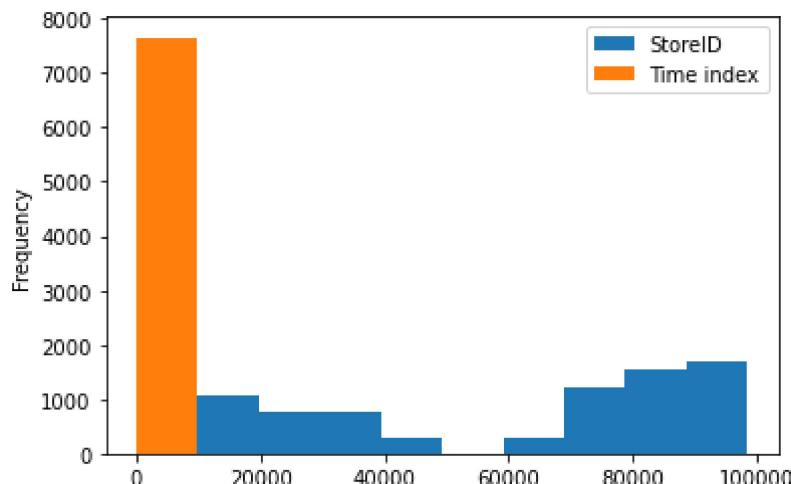
Out[13]: <AxesSubplot:>



histogram

In [14]: `df.plot.hist()`

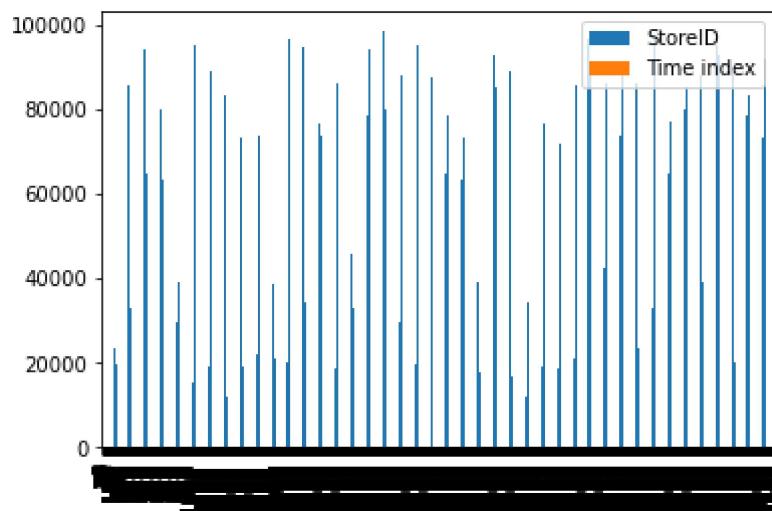
Out[14]: <AxesSubplot:ylabel='Frequency'>



bar chart

In [15]: `df.plot.bar()`

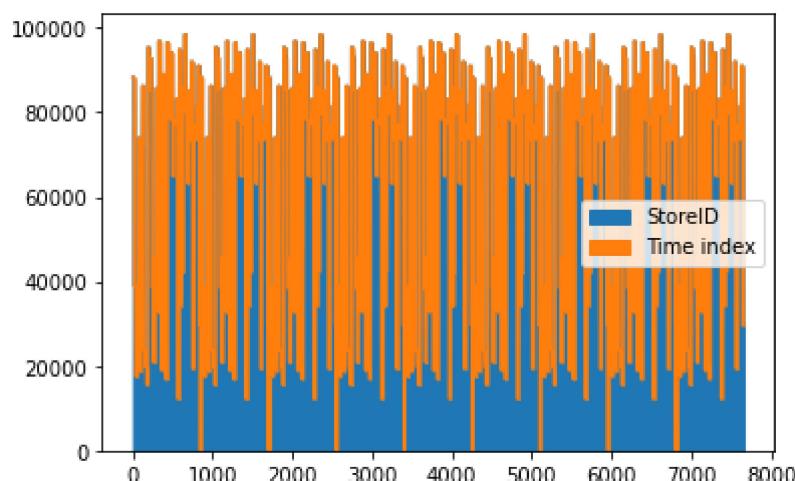
Out[15]: <AxesSubplot:>



area plot

In [16]: `df.plot.area()`

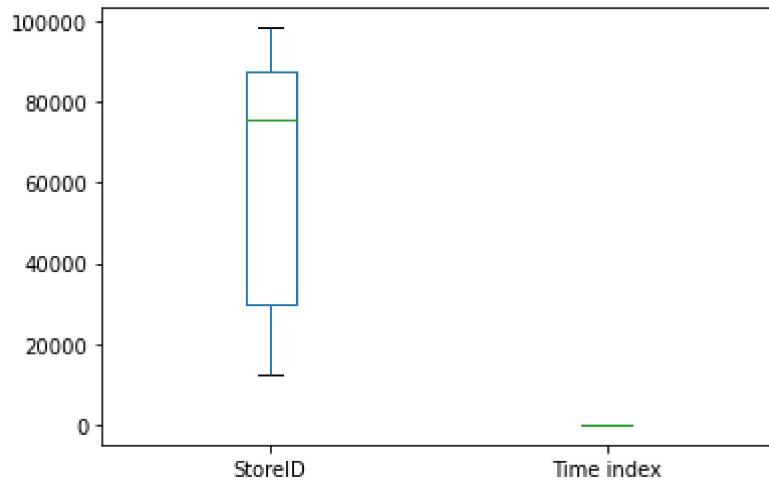
Out[16]: <AxesSubplot:>



box plot

In [17]: `df.plot.box()`

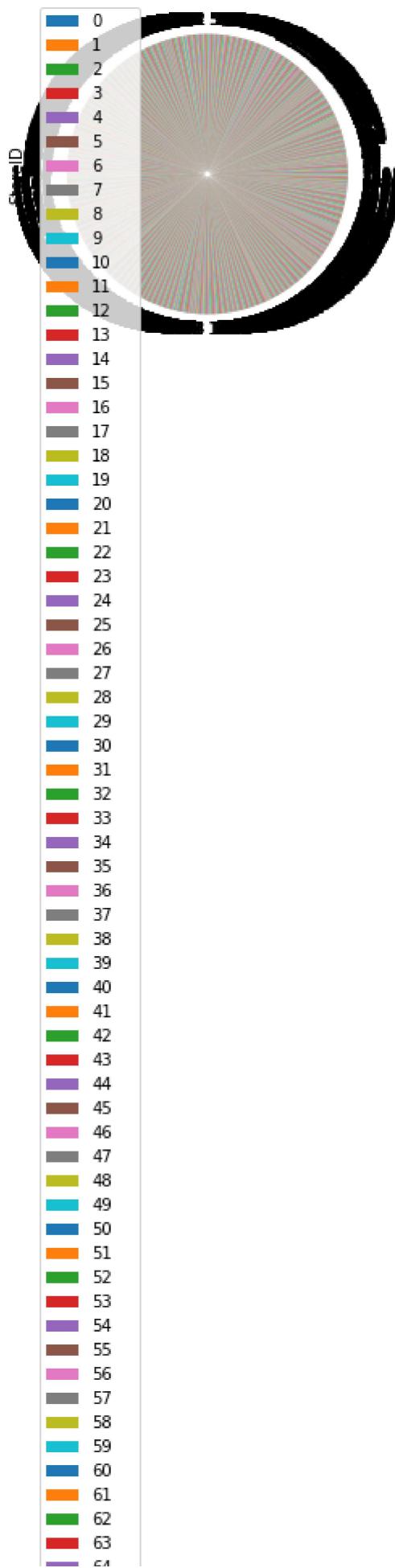
Out[17]: <AxesSubplot:>



pie plot

```
In [20]: df1=pd.DataFrame(data[['StoreID','Time index']][0:2000])  
df1.plot.pie(y="StoreID")
```

```
Out[20]: <AxesSubplot:ylabel='StoreID'>
```

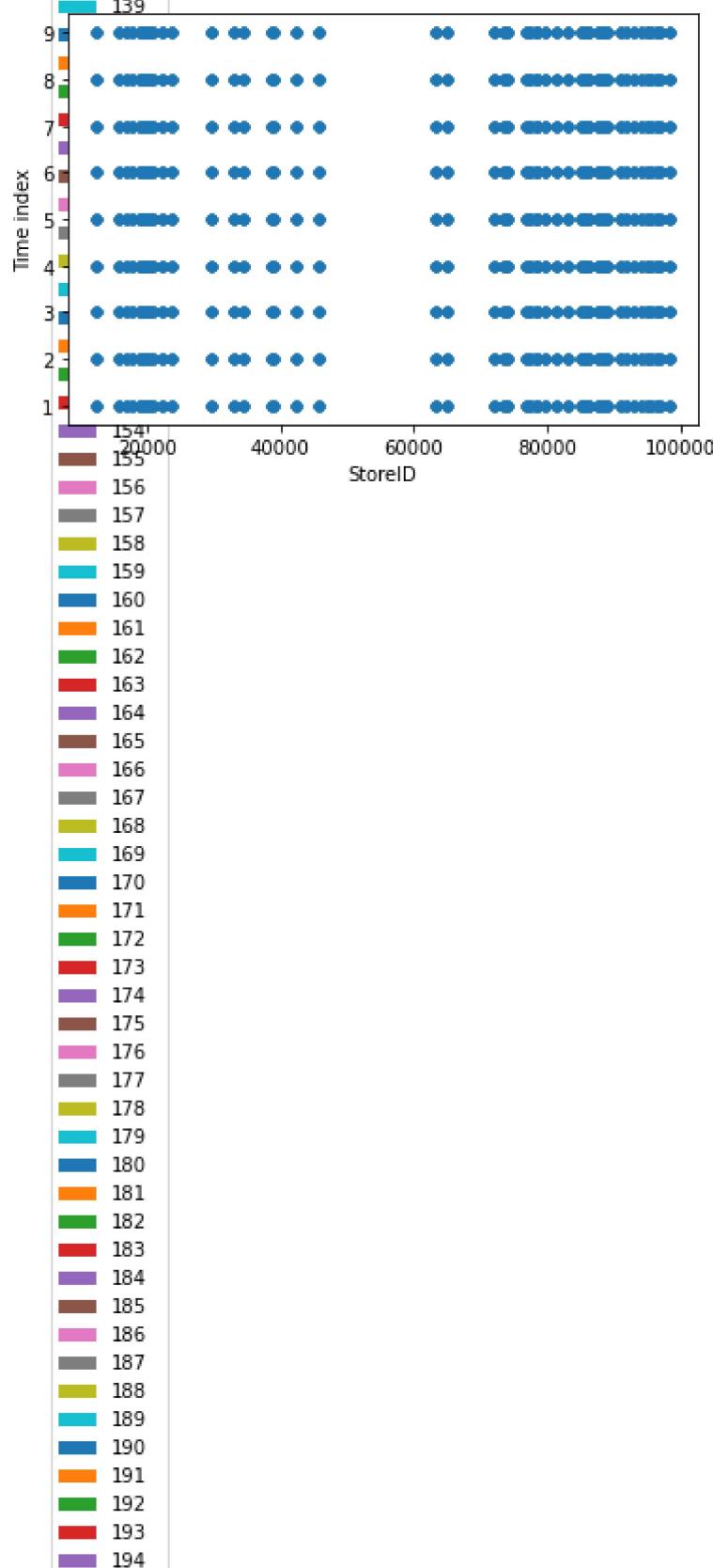


64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

scatter plot

```
In [19]: df.plot.scatter(x="StoreID",y="Time index")
```

```
Out[19]: <AxesSubplot:xlabel='StoreID', ylabel='Time index'>
```



195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260

260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325

326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390

391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

450
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521

522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586

587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652

652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717

718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782

783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848

840
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913

914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978

979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044

1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109

■ 1110
■ 1111
■ 1112
■ 1113
■ 1114
■ 1115
■ 1116
■ 1117
■ 1118
■ 1119
■ 1120
■ 1121
■ 1122
■ 1123
■ 1124
■ 1125
■ 1126
■ 1127
■ 1128
■ 1129
■ 1130
■ 1131
■ 1132
■ 1133
■ 1134
■ 1135
■ 1136
■ 1137
■ 1138
■ 1139
■ 1140
■ 1141
■ 1142
■ 1143
■ 1144
■ 1145
■ 1146
■ 1147
■ 1148
■ 1149
■ 1150
■ 1151
■ 1152
■ 1153
■ 1154
■ 1155
■ 1156
■ 1157
■ 1158
■ 1159
■ 1160
■ 1161
■ 1162
■ 1163
■ 1164
■ 1165
■ 1166
■ 1167
■ 1168
■ 1169
■ 1170
■ 1171
■ 1172
■ 1173
■ 1174

■	1175
■	1176
■	1177
■	1178
■	1179
■	1180
■	1181
■	1182
■	1183
■	1184
■	1185
■	1186
■	1187
■	1188
■	1189
■	1190
■	1191
■	1192
■	1193
■	1194
■	1195
■	1196
■	1197
■	1198
■	1199
■	1200
■	1201
■	1202
■	1203
■	1204
■	1205
■	1206
■	1207
■	1208
■	1209
■	1210
■	1211
■	1212
■	1213
■	1214
■	1215
■	1216
■	1217
■	1218
■	1219
■	1220
■	1221
■	1222
■	1223
■	1224
■	1225
■	1226
■	1227
■	1228
■	1229
■	1230
■	1231
■	1232
■	1233
■	1234
■	1235
■	1236
■	1237
■	1238
■	1239
■	1240

1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305

■ 1306
■ 1307
■ 1308
■ 1309
■ 1310
■ 1311
■ 1312
■ 1313
■ 1314
■ 1315
■ 1316
■ 1317
■ 1318
■ 1319
■ 1320
■ 1321
■ 1322
■ 1323
■ 1324
■ 1325
■ 1326
■ 1327
■ 1328
■ 1329
■ 1330
■ 1331
■ 1332
■ 1333
■ 1334
■ 1335
■ 1336
■ 1337
■ 1338
■ 1339
■ 1340
■ 1341
■ 1342
■ 1343
■ 1344
■ 1345
■ 1346
■ 1347
■ 1348
■ 1349
■ 1350
■ 1351
■ 1352
■ 1353
■ 1354
■ 1355
■ 1356
■ 1357
■ 1358
■ 1359
■ 1360
■ 1361
■ 1362
■ 1363
■ 1364
■ 1365
■ 1366
■ 1367
■ 1368
■ 1369
■ 1370

1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436

■ 1438
■ 1439
■ 1440
■ 1441
■ 1442
■ 1443
■ 1444
■ 1445
■ 1446
■ 1447
■ 1448
■ 1449
■ 1450
■ 1451
■ 1452
■ 1453
■ 1454
■ 1455
■ 1456
■ 1457
■ 1458
■ 1459
■ 1460
■ 1461
■ 1462
■ 1463
■ 1464
■ 1465
■ 1466
■ 1467
■ 1468
■ 1469
■ 1470
■ 1471
■ 1472
■ 1473
■ 1474
■ 1475
■ 1476
■ 1477
■ 1478
■ 1479
■ 1480
■ 1481
■ 1482
■ 1483
■ 1484
■ 1485
■ 1486
■ 1487
■ 1488
■ 1489
■ 1490
■ 1491
■ 1492
■ 1493
■ 1494
■ 1495
■ 1496
■ 1497
■ 1498
■ 1499
■ 1500
■ 1501

1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566

■	1567
■	1568
■	1569
■	1570
■	1571
■	1572
■	1573
■	1574
■	1575
■	1576
■	1577
■	1578
■	1579
■	1580
■	1581
■	1582
■	1583
■	1584
■	1585
■	1586
■	1587
■	1588
■	1589
■	1590
■	1591
■	1592
■	1593
■	1594
■	1595
■	1596
■	1597
■	1598
■	1599
■	1600
■	1601
■	1602
■	1603
■	1604
■	1605
■	1606
■	1607
■	1608
■	1609
■	1610
■	1611
■	1612
■	1613
■	1614
■	1615
■	1616
■	1617
■	1618
■	1619
■	1620
■	1621
■	1622
■	1623
■	1624
■	1625
■	1626
■	1627
■	1628
■	1629
■	1630
■	1631
■	1632

1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697

1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762

1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828

1820
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893

1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958

