Task 1: Define a function called hypotenuse that calculates the length of the hypotenuse of a right triangle when the other two sides are given. Use this function in a program to determine the length of the hypotenuse for each of the following triangles. The function should take twoarguments of float type and return the hypotenuse as float type too. The sample output is as following

In [2]:
```
def pyth(a,b):
    c = (a**2+b**2)**0.5;

    return c

y=float(pyth(6,7))
print("The length of hyp is: ",y,"m")
```

```
The length of hyp is:  9.219544457292887 m
```

Task 2: Write a function distance that calculates the distance between two points (x1, y1) and (x2, y2). All numbers and return values should be of floating type. Use this function in your program. The sample output is as following:

In [9]:
```
import math

def distance (x1, y1, x2, y2):
    distance = math.sqrt((x2-x1)**2 + (y2-y1)**2)
    return distance
point1 = (2,4)
point2 = (5,6)
result = distance(point1[0],point1[1],point2[0],point2[1])
print(f"The distance between {point1} and {point2} is {result:.2f}")
```

```
The distance between (2, 4) and (5, 6) is 3.61
```

Task 3: Write a function reversed that takes an integer value (between 1-9999) and returns the number with its digits reversed. For example, given the number 7631, the function should return 1367. Demonstrate the use of this function in your program with the following sample output.

In [10]:
```
def reversed_number(num):
    if 1 <= num <= 9999:
        reversed_num = int(str(num)[::-1])
        return reversed_num
    else:
        return "Number should be between 1 and 9999"

num = 7631
result = reversed_number(num)
print(f"The reversed number of {num} is {result}")
```

```
The reversed number of 7631 is 1367
```

Task 4: Write a function is_prime that accepts an integer as argument and returns True if the number is prime and False otherwise, Use this function in your program to print on screen all the 4 digit prime numbers.

In [11]:
```
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True
for num in range(1000, 10000):
    if is_prime(num):
        print(num)
```

1009
1013
1019
1021
1031
1033
1039
1049
1051
1061
1063
1069
1087
1091
1093
1097
1103
1109
1117
1123
1129
1151
1153
1163
1171
1181
1187
1193
1201
1213
1217
1223
1229
1231
1237
1249
1259
1277
1279
1283
1289
1291
1297
1301
1303
1307
1319
1321
1327
1361
1367
1373
1381
1399
1409
1423
1427
1429
1433
1439
1447
1451
1453
1459

1471
1481
1483
1487
1489
1493
1499
1511
1523
1531
1543
1549
1553
1559
1567
1571
1579
1583
1597
1601
1607
1609
1613
1619
1621
1627
1637
1657
1663
1667
1669
1693
1697
1699
1709
1721
1723
1733
1741
1747
1753
1759
1777
1783
1787
1789
1801
1811
1823
1831
1847
1861
1867
1871
1873
1877
1879
1889
1901
1907
1913
1931
1933
1949

1951
1973
1979
1987
1993
1997
1999
2003
2011
2017
2027
2029
2039
2053
2063
2069
2081
2083
2087
2089
2099
2111
2113
2129
2131
2137
2141
2143
2153
2161
2179
2203
2207
2213
2221
2237
2239
2243
2251
2267
2269
2273
2281
2287
2293
2297
2309
2311
2333
2339
2341
2347
2351
2357
2371
2377
2381
2383
2389
2393
2399
2411
2417
2423

2437
2441
2447
2459
2467
2473
2477
2503
2521
2531
2539
2543
2549
2551
2557
2579
2591
2593
2609
2617
2621
2633
2647
2657
2659
2663
2671
2677
2683
2687
2689
2693
2699
2707
2711
2713
2719
2729
2731
2741
2749
2753
2767
2777
2789
2791
2797
2801
2803
2819
2833
2837
2843
2851
2857
2861
2879
2887
2897
2903
2909
2917
2927
2939

2953
2957
2963
2969
2971
2999
3001
3011
3019
3023
3037
3041
3049
3061
3067
3079
3083
3089
3109
3119
3121
3137
3163
3167
3169
3181
3187
3191
3203
3209
3217
3221
3229
3251
3253
3257
3259
3271
3299
3301
3307
3313
3319
3323
3329
3331
3343
3347
3359
3361
3371
3373
3389
3391
3407
3413
3433
3449
3457
3461
3463
3467
3469
3491

3499
3511
3517
3527
3529
3533
3539
3541
3547
3557
3559
3571
3581
3583
3593
3607
3613
3617
3623
3631
3637
3643
3659
3671
3673
3677
3691
3697
3701
3709
3719
3727
3733
3739
3761
3767
3769
3779
3793
3797
3803
3821
3823
3833
3847
3851
3853
3863
3877
3881
3889
3907
3911
3917
3919
3923
3929
3931
3943
3947
3967
3989
4001
4003

4007
4013
4019
4021
4027
4049
4051
4057
4073
4079
4091
4093
4099
4111
4127
4129
4133
4139
4153
4157
4159
4177
4201
4211
4217
4219
4229
4231
4241
4243
4253
4259
4261
4271
4273
4283
4289
4297
4327
4337
4339
4349
4357
4363
4373
4391
4397
4409
4421
4423
4441
4447
4451
4457
4463
4481
4483
4493
4507
4513
4517
4519
4523
4547

4549
4561
4567
4583
4591
4597
4603
4621
4637
4639
4643
4649
4651
4657
4663
4673
4679
4691
4703
4721
4723
4729
4733
4751
4759
4783
4787
4789
4793
4799
4801
4813
4817
4831
4861
4871
4877
4889
4903
4909
4919
4931
4933
4937
4943
4951
4957
4967
4969
4973
4987
4993
4999
5003
5009
5011
5021
5023
5039
5051
5059
5077
5081
5087

5099
5101
5107
5113
5119
5147
5153
5167
5171
5179
5189
5197
5209
5227
5231
5233
5237
5261
5273
5279
5281
5297
5303
5309
5323
5333
5347
5351
5381
5387
5393
5399
5407
5413
5417
5419
5431
5437
5441
5443
5449
5471
5477
5479
5483
5501
5503
5507
5519
5521
5527
5531
5557
5563
5569
5573
5581
5591
5623
5639
5641
5647
5651
5653

5657
5659
5669
5683
5689
5693
5701
5711
5717
5737
5741
5743
5749
5779
5783
5791
5801
5807
5813
5821
5827
5839
5843
5849
5851
5857
5861
5867
5869
5879
5881
5897
5903
5923
5927
5939
5953
5981
5987
6007
6011
6029
6037
6043
6047
6053
6067
6073
6079
6089
6091
6101
6113
6121
6131
6133
6143
6151
6163
6173
6197
6199
6203
6211

6217
6221
6229
6247
6257
6263
6269
6271
6277
6287
6299
6301
6311
6317
6323
6329
6337
6343
6353
6359
6361
6367
6373
6379
6389
6397
6421
6427
6449
6451
6469
6473
6481
6491
6521
6529
6547
6551
6553
6563
6569
6571
6577
6581
6599
6607
6619
6637
6653
6659
6661
6673
6679
6689
6691
6701
6703
6709
6719
6733
6737
6761
6763
6779

6781
6791
6793
6803
6823
6827
6829
6833
6841
6857
6863
6869
6871
6883
6899
6907
6911
6917
6947
6949
6959
6961
6967
6971
6977
6983
6991
6997
7001
7013
7019
7027
7039
7043
7057
7069
7079
7103
7109
7121
7127
7129
7151
7159
7177
7187
7193
7207
7211
7213
7219
7229
7237
7243
7247
7253
7283
7297
7307
7309
7321
7331
7333
7349

7351
7369
7393
7411
7417
7433
7451
7457
7459
7477
7481
7487
7489
7499
7507
7517
7523
7529
7537
7541
7547
7549
7559
7561
7573
7577
7583
7589
7591
7603
7607
7621
7639
7643
7649
7669
7673
7681
7687
7691
7699
7703
7717
7723
7727
7741
7753
7757
7759
7789
7793
7817
7823
7829
7841
7853
7867
7873
7877
7879
7883
7901
7907
7919

7927
7933
7937
7949
7951
7963
7993
8009
8011
8017
8039
8053
8059
8069
8081
8087
8089
8093
8101
8111
8117
8123
8147
8161
8167
8171
8179
8191
8209
8219
8221
8231
8233
8237
8243
8263
8269
8273
8287
8291
8293
8297
8311
8317
8329
8353
8363
8369
8377
8387
8389
8419
8423
8429
8431
8443
8447
8461
8467
8501
8513
8521
8527
8537

8539
8543
8563
8573
8581
8597
8599
8609
8623
8627
8629
8641
8647
8663
8669
8677
8681
8689
8693
8699
8707
8713
8719
8731
8737
8741
8747
8753
8761
8779
8783
8803
8807
8819
8821
8831
8837
8839
8849
8861
8863
8867
8887
8893
8923
8929
8933
8941
8951
8963
8969
8971
8999
9001
9007
9011
9013
9029
9041
9043
9049
9059
9067
9091

9103
9109
9127
9133
9137
9151
9157
9161
9173
9181
9187
9199
9203
9209
9221
9227
9239
9241
9257
9277
9281
9283
9293
9311
9319
9323
9337
9341
9343
9349
9371
9377
9391
9397
9403
9413
9419
9421
9431
9433
9437
9439
9461
9463
9467
9473
9479
9491
9497
9511
9521
9533
9539
9547
9551
9587
9601
9613
9619
9623
9629
9631
9643
9649

```
9661
9677
9679
9689
9697
9719
9721
9733
9739
9743
9749
9767
9769
9781
9787
9791
9803
9811
9817
9829
9833
9839
9851
9857
9859
9871
9883
9887
9901
9907
9923
9929
9931
9941
9949
9967
9973
```

Task 5: Write a function qualityPoints that inputs a student's average and returns 4 if a student's average is 90-100, 3 if the average is 80-89, 2 if the average is 70-79, 1 if the average is 60-69, and 0 if the average is lower than 60. Use this function in your program that reads average from the student and prints on screen the corresponding GPA as shown below.

In [14]:
```python
def qualityPoints(average):
    if average >= 90 and average <= 100:
        return 4
    elif average >= 80 and average <= 89:
        return 3
    elif average >= 70 and average <= 79:
        return 2
    elif average >= 60 and average <= 69:
        return 1
```

```
        return 0

# Example usage
average = float(input("Enter student's average: "))
gpa = qualityPoints(average)
print(f"The corresponding GPA is: {gpa}")
```

```
Enter student's average: 80
The corresponding GPA is: 3
```

Task 6: Write a function is_bouncy that accepts an integer as argument and returns True if the number is bouncy and False otherwise. Use this function in your program to print on screen all the 4 digit bouncy numbers along with the total count of the 4-digit bouncy numbers.

In [1]:
```
def is_bouncy(num):
    num_str = str(num)
    increasing = decreasing = False

    for i in range(1, len(num_str)):
        if num_str[i] > num_str[i - 1]:
            increasing = True
        elif num_str[i] < num_str[i - 1]:
            decreasing = True

        if increasing and decreasing:
            return True

    return False

def find_4_digit_bouncy_numbers():
    count = 0
    for num in range(1000, 10000):
        if is_bouncy(num):
            print(num)
            count += 1

    return count

# Call the function to find and print 4-digit bouncy numbers
count = find_4_digit_bouncy_numbers()
print("Total count of 4-digit bouncy numbers:", count)
```

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
1101
1102
1103
1104
1105
1106
1107
1108
1109
1120
1121
1130
1131
1132
1140
1141
1142
1143
1150
1151
1152
1153
1154
1160
1161
1162
1163
1164
1165

1170
1171
1172
1173
1174
1175
1176
1180
1181
1182
1183
1184
1185
1186
1187
1190
1191
1192
1193
1194
1195
1196
1197
1198
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1230
1231
1232
1240
1241
1242
1243
1250
1251
1252
1253
1254
1260
1261
1262
1263
1264
1265

1270
1271
1272
1273
1274
1275
1276
1280
1281
1282
1283
1284
1285
1286
1287
1290
1291
1292
1293
1294
1295
1296
1297
1298
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
1340
1341
1342
1343
1350
1351
1352

1353
1354
1360
1361
1362
1363
1364
1365
1370
1371
1372
1373
1374
1375
1376
1380
1381
1382
1383
1384
1385
1386
1387
1390
1391
1392
1393
1394
1395
1396
1397
1398
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
1437
1438
1439
1440
1441
1442
1443
1450
1451
1452
1453
1454
1460
1461
1462
1463
1464
1465
1470
1471
1472
1473
1474
1475
1476
1480
1481
1482
1483
1484
1485
1486
1487
1490
1491
1492
1493
1494
1495
1496
1497
1498
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
1560
1561
1562
1563
1564
1565
1570
1571
1572
1573
1574
1575
1576
1580
1581
1582
1583
1584
1585
1586
1587
1590
1591
1592
1593
1594

1595
1596
1597
1598
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
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
1670
1671
1672
1673
1674
1675
1676
1680
1681
1682
1683
1684
1685
1686
1687
1690
1691
1692
1693
1694
1695
1696
1697
1698
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
1780
1781
1782
1783
1784
1785
1786
1787
1790
1791
1792
1793
1794
1795
1796
1797
1798
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
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
1890
1891
1892
1893
1894
1895
1896
1897
1898
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
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998

2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064

2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2101
2102
2103
2104
2105
2106
2107
2108
2109
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131

2196
2197
2198
2199
2201
2202
2203
2204
2205
2206
2207
2208
2209
2212
2213
2214
2215
2216
2217
2218
2219
2230
2231
2232
2240
2241
2242
2243
2250
2251
2252
2253
2254
2260
2261
2262
2263
2264
2265
2270
2271
2272
2273
2274
2275
2276
2280
2281
2282
2283
2284
2285
2286
2287
2290
2291
2292
2293
2294
2295
2296
2297
2298
2300

2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2340
2341
2342
2343
2350
2351
2352
2353
2354
2360
2361
2362
2363
2364
2365
2370
2371
2372
2373
2374
2375
2376
2380
2381
2382
2383
2384
2385
2386
2387
2390
2391

2392
2393
2394
2395
2396
2397
2398
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2450
2451
2452
2453
2454
2460
2461
2462
2463
2464
2465
2470
2471

2472
2473
2474
2475
2476
2480
2481
2482
2483
2484
2485
2486
2487
2490
2491
2492
2493
2494
2495
2496
2497
2498
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541

2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2560
2561
2562
2563
2564
2565
2570
2571
2572
2573
2574
2575
2576
2580
2581
2582
2583
2584
2585
2586
2587
2590
2591
2592
2593
2594
2595
2596
2597
2598
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620

2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2670
2671
2672
2673
2674
2675
2676
2680
2681
2682
2683
2684
2685
2686
2687
2690
2691
2692
2693

2694
2695
2696
2697
2698
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758

2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2780
2781
2782
2783
2784
2785
2786
2787
2790
2791
2792
2793
2794
2795
2796
2797
2798
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828

2895
2896
2897
2898
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959

2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025

3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3101
3102
3103
3104
3105
3106
3107
3108
3109
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156

3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3201
3202
3203
3204
3205
3206
3207
3208
3209
3212
3213
3214
3215
3216
3217
3218
3219
3223
3224
3225
3226

3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290

3291
3292
3293
3294
3295
3296
3297
3298
3299
3301
3302
3303
3304
3305
3306
3307
3308
3309
3312
3313
3314
3315
3316
3317
3318
3319
3323
3324
3325
3326
3327
3328
3329
3340
3341
3342
3343
3350
3351
3352
3353
3354
3360
3361
3362
3363
3364
3365
3370
3371
3372
3373
3374
3375
3376
3380
3381
3382
3383
3384
3385
3386
3387
3390

3391
3392
3393
3394
3395
3396
3397
3398
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3450
3451
3452
3453
3454
3460
3461
3462
3463
3464
3465
3470

3471
3472
3473
3474
3475
3476
3480
3481
3482
3483
3484
3485
3486
3487
3490
3491
3492
3493
3494
3495
3496
3497
3498
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540

3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3560
3561
3562
3563
3564
3565
3570
3571
3572
3573
3574
3575
3576
3580
3581
3582
3583
3584
3585
3586
3587
3590
3591
3592
3593
3594
3595
3596
3597
3598
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619

3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3670
3671
3672
3673
3674
3675
3676
3680
3681
3682
3683
3684
3685
3686
3687
3690
3691
3692

3693
3694
3695
3696
3697
3698
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757

3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3780
3781
3782
3783
3784
3785
3786
3787
3790
3791
3792
3793
3794
3795
3796
3797
3798
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827

3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3890
3891
3892
3893

3894
3895
3896
3897
3898
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958

3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024

4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088

4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4101
4102
4103
4104
4105
4106
4107
4108
4109
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155

4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4201
4202
4203
4204
4205
4206
4207
4208
4209
4212
4213
4214
4215
4216
4217
4218
4219
4223
4224
4225

4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289

4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4301
4302
4303
4304
4305
4306
4307
4308
4309
4312
4313
4314
4315
4316
4317
4318
4319
4323
4324
4325
4326
4327
4328
4329
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363

4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4401
4402
4403
4404
4405
4406
4407
4408
4409
4412
4413
4414
4415
4416
4417
4418
4419
4423
4424
4425
4426
4427
4428
4429
4434
4435
4436
4437

4438
4439
4450
4451
4452
4453
4454
4460
4461
4462
4463
4464
4465
4470
4471
4472
4473
4474
4475
4476
4480
4481
4482
4483
4484
4485
4486
4487
4490
4491
4492
4493
4494
4495
4496
4497
4498
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526

4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4560
4561
4562
4563
4564
4565
4570
4571
4572
4573
4574
4575
4576
4580
4581
4582
4583
4584
4585
4586
4587
4590
4591
4592
4593
4594
4595
4596
4597
4598
4600
4601
4602
4603
4604
4605

4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4670
4671
4672
4673

4674
4675
4676
4680
4681
4682
4683
4684
4685
4686
4687
4690
4691
4692
4693
4694
4695
4696
4697
4698
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743

4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4780
4781
4782
4783
4784
4785
4786
4787
4790
4791
4792
4793
4794
4795
4796
4797
4798
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
4890
4891
4892
4893
4894
4895
4896
4897
4898
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
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
5101
5102
5103
5104
5105
5106
5107
5108
5109
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
5201
5202
5203
5204
5205
5206

5207
5208
5209
5212
5213
5214
5215
5216
5217
5218
5219
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
5301
5302
5303
5304
5305
5306
5307
5308
5309
5312
5313
5314
5315
5316
5317
5318
5319
5323
5324
5325
5326
5327
5328
5329
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
5401
5402
5403
5404
5405
5406
5407
5408
5409
5412
5413
5414
5415
5416

5417
5418
5419
5423
5424
5425
5426
5427
5428
5429
5434
5435
5436
5437
5438
5439
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
5501
5502
5503
5504
5505
5506
5507
5508
5509
5512
5513
5514
5515
5516
5517
5518
5519
5523
5524
5525
5526
5527
5528
5529
5534
5535
5536
5537
5538
5539
5545
5546
5547
5548
5549
5560
5561
5562
5563
5564
5565
5570
5571
5572
5573
5574
5575
5576
5580
5581
5582
5583
5584
5585
5586
5587
5590

5591
5592
5593
5594
5595
5596
5597
5598
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
5670
5671
5672
5673
5674
5675
5676
5680
5681
5682
5683
5684
5685
5686
5687
5690
5691
5692
5693
5694
5695
5696
5697
5698
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729

5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5780
5781
5782
5783
5784
5785
5786
5787
5790
5791
5792
5793
5794
5795
5796
5797
5798

5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863

5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5890
5891
5892
5893
5894
5895
5896
5897
5898
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930

5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994

5995
5996
5997
5998
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060

6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6101
6102
6103
6104
6105
6106
6107
6108
6109
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127

6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191

6192
6193
6194
6195
6196
6197
6198
6199
6201
6202
6203
6204
6205
6206
6207
6208
6209
6212
6213
6214
6215
6216
6217
6218
6219
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261

6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6301
6302
6303
6304
6305
6306
6307
6308
6309
6312
6313
6314
6315
6316
6317
6318
6319
6323
6324
6325
6326
6327
6328
6329
6334
6335

6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399

6401
6402
6403
6404
6405
6406
6407
6408
6409
6412
6413
6414
6415
6416
6417
6418
6419
6423
6424
6425
6426
6427
6428
6429
6434
6435
6436
6437
6438
6439
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478

6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6501
6502
6503
6504
6505
6506
6507
6508
6509
6512
6513
6514
6515
6516
6517
6518
6519
6523
6524
6525
6526
6527
6528
6529
6534
6535
6536
6537
6538
6539
6545
6546
6547
6548
6549
6556
6557
6558
6559
6560
6561
6562
6563

6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6601
6602
6603
6604
6605
6606
6607
6608
6609
6612
6613
6614
6615
6616
6617
6618
6619
6623
6624
6625
6626
6627
6628
6629
6634
6635
6636
6637

6638
6639
6645
6646
6647
6648
6649
6656
6657
6658
6659
6670
6671
6672
6673
6674
6675
6676
6680
6681
6682
6683
6684
6685
6686
6687
6690
6691
6692
6693
6694
6695
6696
6697
6698
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728

6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6780
6781
6782
6783
6784
6785
6786
6787
6790
6791
6792
6793
6794
6795
6796
6797

6798
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862

6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6890
6891
6892
6893
6894
6895
6896
6897
6898
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929

6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993

6994
6995
6996
6997
6998
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059

7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7101
7102
7103
7104
7105
7106
7107
7108
7109
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126

7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190

7191
7192
7193
7194
7195
7196
7197
7198
7199
7201
7202
7203
7204
7205
7206
7207
7208
7209
7212
7213
7214
7215
7216
7217
7218
7219
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235
7236
7237
7238
7239
7240
7241
7242
7243
7244
7245
7246
7247
7248
7249
7250
7251
7252
7253
7254
7255
7256
7257
7258
7259
7260

7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7301
7302
7303
7304
7305
7306
7307
7308
7309
7312
7313
7314
7315
7316
7317
7318
7319
7323
7324
7325
7326
7327
7328
7329
7334

7335
7336
7337
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358
7359
7360
7361
7362
7363
7364
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398

7399
7401
7402
7403
7404
7405
7406
7407
7408
7409
7412
7413
7414
7415
7416
7417
7418
7419
7423
7424
7425
7426
7427
7428
7429
7434
7435
7436
7437
7438
7439
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469
7470
7471
7472
7473
7474
7475
7476
7477

7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7501
7502
7503
7504
7505
7506
7507
7508
7509
7512
7513
7514
7515
7516
7517
7518
7519
7523
7524
7525
7526
7527
7528
7529
7534
7535
7536
7537
7538
7539
7545
7546
7547
7548
7549
7556
7557
7558
7559
7560
7561
7562

7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7601
7602
7603
7604
7605
7606
7607
7608
7609
7612
7613
7614
7615
7616
7617
7618
7619
7623
7624
7625
7626
7627
7628
7629
7634
7635
7636

7637
7638
7639
7645
7646
7647
7648
7649
7656
7657
7658
7659
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7701
7702
7703
7704
7705
7706
7707
7708
7709
7712
7713
7714
7715
7716
7717
7718
7719
7723
7724

7725
7726
7727
7728
7729
7734
7735
7736
7737
7738
7739
7745
7746
7747
7748
7749
7756
7757
7758
7759
7767
7768
7769
7780
7781
7782
7783
7784
7785
7786
7787
7790
7791
7792
7793
7794
7795
7796
7797
7798
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823

7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887

7890
7891
7892
7893
7894
7895
7896
7897
7898
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954

7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
8001
8002
8003
8004
8005
8006
8007
8008
8009
8010
8011
8012
8013
8014
8015
8016
8017
8018
8019
8020

8021
8022
8023
8024
8025
8026
8027
8028
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038
8039
8040
8041
8042
8043
8044
8045
8046
8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083
8084

8085
8086
8087
8088
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8099
8101
8102
8103
8104
8105
8106
8107
8108
8109
8112
8113
8114
8115
8116
8117
8118
8119
8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149
8150
8151

8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8170
8171
8172
8173
8174
8175
8176
8177
8178
8179
8180
8181
8182
8183
8184
8185
8186
8187
8188
8189
8190
8191
8192
8193
8194
8195
8196
8197
8198
8199
8201
8202
8203
8204
8205
8206
8207
8208
8209
8212
8213
8214
8215
8216
8217
8218

8219
8223
8224
8225
8226
8227
8228
8229
8230
8231
8232
8233
8234
8235
8236
8237
8238
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249
8250
8251
8252
8253
8254
8255
8256
8257
8258
8259
8260
8261
8262
8263
8264
8265
8266
8267
8268
8269
8270
8271
8272
8273
8274
8275
8276
8277
8278
8279
8280
8281
8282
8283
8284
8285

8286
8287
8288
8289
8290
8291
8292
8293
8294
8295
8296
8297
8298
8299
8301
8302
8303
8304
8305
8306
8307
8308
8309
8312
8313
8314
8315
8316
8317
8318
8319
8323
8324
8325
8326
8327
8328
8329
8334
8335
8336
8337
8338
8339
8340
8341
8342
8343
8344
8345
8346
8347
8348
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359

8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8401
8402
8403
8404
8405
8406
8407
8408
8409
8412
8413
8414
8415
8416
8417
8418
8419
8423
8424
8425
8426
8427
8428
8429

8434
8435
8436
8437
8438
8439
8445
8446
8447
8448
8449
8450
8451
8452
8453
8454
8455
8456
8457
8458
8459
8460
8461
8462
8463
8464
8465
8466
8467
8468
8469
8470
8471
8472
8473
8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484
8485
8486
8487
8488
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8499
8501
8502
8503

8504
8505
8506
8507
8508
8509
8512
8513
8514
8515
8516
8517
8518
8519
8523
8524
8525
8526
8527
8528
8529
8534
8535
8536
8537
8538
8539
8545
8546
8547
8548
8549
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8579
8580
8581
8582
8583
8584
8585
8586
8587

8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599
8601
8602
8603
8604
8605
8606
8607
8608
8609
8612
8613
8614
8615
8616
8617
8618
8619
8623
8624
8625
8626
8627
8628
8629
8634
8635
8636
8637
8638
8639
8645
8646
8647
8648
8649
8656
8657
8658
8659
8667
8668
8669
8670
8671
8672
8673
8674
8675
8676
8677
8678
8679

8680
8681
8682
8683
8684
8685
8686
8687
8688
8689
8690
8691
8692
8693
8694
8695
8696
8697
8698
8699
8701
8702
8703
8704
8705
8706
8707
8708
8709
8712
8713
8714
8715
8716
8717
8718
8719
8723
8724
8725
8726
8727
8728
8729
8734
8735
8736
8737
8738
8739
8745
8746
8747
8748
8749
8756
8757
8758
8759
8767
8768
8769
8778
8779

8780
8781
8782
8783
8784
8785
8786
8787
8788
8789
8790
8791
8792
8793
8794
8795
8796
8797
8798
8799
8801
8802
8803
8804
8805
8806
8807
8808
8809
8812
8813
8814
8815
8816
8817
8818
8819
8823
8824
8825
8826
8827
8828
8829
8834
8835
8836
8837
8838
8839
8845
8846
8847
8848
8849
8856
8857
8858
8859
8867
8868
8869
8878
8879

8890
8891
8892
8893
8894
8895
8896
8897
8898
8900
8901
8902
8903
8904
8905
8906
8907
8908
8909
8910
8911
8912
8913
8914
8915
8916
8917
8918
8919
8920
8921
8922
8923
8924
8925
8926
8927
8928
8929
8930
8931
8932
8933
8934
8935
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945
8946
8947
8948
8949
8950
8951
8952
8953
8954

8955
8956
8957
8958
8959
8960
8961
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983
8984
8985
8986
8987
8988
8989
8990
8991
8992
8993
8994
8995
8996
8997
8998
9001
9002
9003
9004
9005
9006
9007
9008
9009
9010
9011
9012
9013
9014
9015
9016
9017
9018
9019
9020

9021
9022
9023
9024
9025
9026
9027
9028
9029
9030
9031
9032
9033
9034
9035
9036
9037
9038
9039
9040
9041
9042
9043
9044
9045
9046
9047
9048
9049
9050
9051
9052
9053
9054
9055
9056
9057
9058
9059
9060
9061
9062
9063
9064
9065
9066
9067
9068
9069
9070
9071
9072
9073
9074
9075
9076
9077
9078
9079
9080
9081
9082
9083
9084

9085
9086
9087
9088
9089
9090
9091
9092
9093
9094
9095
9096
9097
9098
9099
9101
9102
9103
9104
9105
9106
9107
9108
9109
9112
9113
9114
9115
9116
9117
9118
9119
9120
9121
9122
9123
9124
9125
9126
9127
9128
9129
9130
9131
9132
9133
9134
9135
9136
9137
9138
9139
9140
9141
9142
9143
9144
9145
9146
9147
9148
9149
9150
9151

9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162
9163
9164
9165
9166
9167
9168
9169
9170
9171
9172
9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186
9187
9188
9189
9190
9191
9192
9193
9194
9195
9196
9197
9198
9199
9201
9202
9203
9204
9205
9206
9207
9208
9209
9212
9213
9214
9215
9216
9217
9218

9219
9223
9224
9225
9226
9227
9228
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238
9239
9240
9241
9242
9243
9244
9245
9246
9247
9248
9249
9250
9251
9252
9253
9254
9255
9256
9257
9258
9259
9260
9261
9262
9263
9264
9265
9266
9267
9268
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283
9284
9285

9286
9287
9288
9289
9290
9291
9292
9293
9294
9295
9296
9297
9298
9299
9301
9302
9303
9304
9305
9306
9307
9308
9309
9312
9313
9314
9315
9316
9317
9318
9319
9323
9324
9325
9326
9327
9328
9329
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359

9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385
9386
9387
9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9401
9402
9403
9404
9405
9406
9407
9408
9409
9412
9413
9414
9415
9416
9417
9418
9419
9423
9424
9425
9426
9427
9428
9429

9434
9435
9436
9437
9438
9439
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471
9472
9473
9474
9475
9476
9477
9478
9479
9480
9481
9482
9483
9484
9485
9486
9487
9488
9489
9490
9491
9492
9493
9494
9495
9496
9497
9498
9499
9501
9502
9503

9504
9505
9506
9507
9508
9509
9512
9513
9514
9515
9516
9517
9518
9519
9523
9524
9525
9526
9527
9528
9529
9534
9535
9536
9537
9538
9539
9545
9546
9547
9548
9549
9556
9557
9558
9559
9560
9561
9562
9563
9564
9565
9566
9567
9568
9569
9570
9571
9572
9573
9574
9575
9576
9577
9578
9579
9580
9581
9582
9583
9584
9585
9586
9587

9588
9589
9590
9591
9592
9593
9594
9595
9596
9597
9598
9599
9601
9602
9603
9604
9605
9606
9607
9608
9609
9612
9613
9614
9615
9616
9617
9618
9619
9623
9624
9625
9626
9627
9628
9629
9634
9635
9636
9637
9638
9639
9645
9646
9647
9648
9649
9656
9657
9658
9659
9667
9668
9669
9670
9671
9672
9673
9674
9675
9676
9677
9678
9679

9680
9681
9682
9683
9684
9685
9686
9687
9688
9689
9690
9691
9692
9693
9694
9695
9696
9697
9698
9699
9701
9702
9703
9704
9705
9706
9707
9708
9709
9712
9713
9714
9715
9716
9717
9718
9719
9723
9724
9725
9726
9727
9728
9729
9734
9735
9736
9737
9738
9739
9745
9746
9747
9748
9749
9756
9757
9758
9759
9767
9768
9769
9778
9779

9780
9781
9782
9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794
9795
9796
9797
9798
9799
9801
9802
9803
9804
9805
9806
9807
9808
9809
9812
9813
9814
9815
9816
9817
9818
9819
9823
9824
9825
9826
9827
9828
9829
9834
9835
9836
9837
9838
9839
9845
9846
9847
9848
9849
9856
9857
9858
9859
9867
9868
9869
9878
9879

```
9889
9890
9891
9892
9893
9894
9895
9896
9897
9898
9899
9901
9902
9903
9904
9905
9906
9907
9908
9909
9912
9913
9914
9915
9916
9917
9918
9919
9923
9924
9925
9926
9927
9928
9929
9934
9935
9936
9937
9938
9939
9945
9946
9947
9948
9949
9956
9957
9958
9959
9967
9968
9969
9978
9979
9989
Total count of 4-digit bouncy numbers: 7800
```

Task 7: Write a function called number_of_factors that takes an integer and returns how many factors the number has.

In [6]:
```
def number_of_factors(num):
    count = 0
    for i in range(1, num + 1):
        if num % i == 0:
            count += 1
    return count
```

In [7]:
```
num_factors = number_of_factors(12)
print("Number of factors:", num_factors)
```

```
Number of factors: 6
```

Task 8: Write a function called binom that takes two integers n and k and returns the binomial coefficient . The definition is:

In [8]:
```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

def binom(n, k):
    if k > n:
        return 0
    else:
        return factorial(n) // (factorial(k) * factorial(n - k))
```

In [9]:
```
coefficient = binom(5, 2)
print("Binomial coefficient:", coefficient)
```

```
Binomial coefficient: 10
```

Task 9: Write a program that plays the game of “guess the number” as follows: Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 1000. The program then types:

In [10]:
```
import random

def guess_the_number():
    number_to_guess = random.randint(1, 1000)
    attempts = 0

    print("I have selected a number between 1 and 1000. Can you guess it?")
```

```
    while True:
        user_guess = int(input("Enter your guess: "))
        attempts += 1

        if user_guess < number_to_guess:
            print("Too low! Try again.")
        elif user_guess > number_to_guess:
            print("Too high! Try again.")
        else:
            print("Congratulations! You guessed the number in", attempts, "attempts.")
            break

guess_the_number()
```

```
I have selected a number between 1 and 1000. Can you guess it?
Enter your guess: 45
Too low! Try again.
Enter your guess: 65
Too low! Try again.
Enter your guess: 78
Too low! Try again.
Enter your guess: 34
Too low! Try again.
Enter your guess: 100
Too low! Try again.
Enter your guess: 56
Too low! Try again.
Enter your guess: 12
Too low! Try again.
Enter your guess: 500
Too high! Try again.
Enter your guess: 400
Too high! Try again.
Enter your guess: 300
Too low! Try again.
Enter your guess: 350
Too low! Try again.
Enter your guess: 360
Too low! Try again.
Enter your guess: 370
Too low! Try again.
Enter your guess: 380
Too low! Try again.
Enter your guess: 390
Too low! Try again.
Enter your guess: 400
Too high! Try again.
Enter your guess: 391
Too low! Try again.
Enter your guess: 392
Too low! Try again.
Enter your guess: 393
Too low! Try again.
Enter your guess: 394
Too low! Try again.
Enter your guess: 395
Too low! Try again.
Enter your guess: 396
Too low! Try again.
Enter your guess: 397
Congratulations! You guessed the number in 23 attempts.
```

An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because 6 = 1 + 2+ 3. Write a function perfect that determines if the parameter number is a perfect number or not. The function should take an integer argument and returns Boolean True if the integer is perfect and Boolean False otherwise. Use this function in a program that determines and prints all the perfect numbers between 1 and 1000.

In [12]:
```
def perfect(number):
    factors_sum = 0
    for i in range(1, number):
        if number % i == 0:
            factors_sum += i

    return factors_sum == number

def find_perfect_numbers():
    perfect_nums = []
    for num in range(1, 1001):
        if perfect(num):
            perfect_nums.append(num)

    return perfect_nums

perfect_numbers = find_perfect_numbers()

print("Perfect numbers between 1 and 1000:")
for num in perfect_numbers:
    print(num)
```

```
Perfect numbers between 1 and 1000:
6
28
496
```

Write a function int2binary that accepts an integer as argument and returns its equivalent binary number using while loop and a list Use this function in a program that prints on screen the binary equivalent of numbers from 100 till 1000 in a tabular form. [Hint: Use the division-by-2 algorithm]

In [13]:
```
def int2binary(number):
    binary_list = []
    while number > 0:
        y_list.insert(0, number % 2)
```

```
        number //= 2
    return binary_list

# Print the binary equivalent of numbers from 100 to 1000 in a tabular form
print("Number\tBinary")

for num in range(100, 1001):
    binary_num = int2binary(num)
    binary_str = ''.join(str(bit) for bit in binary_num)
    print(f"{num}\t{binary_str}")
```

Number	Binary
100	1100100
101	1100101
102	1100110
103	1100111
104	1101000
105	1101001
106	1101010
107	1101011
108	1101100
109	1101101
110	1101110
111	1101111
112	1110000
113	1110001
114	1110010
115	1110011
116	1110100
117	1110101
118	1110110
119	1110111
120	1111000
121	1111001
122	1111010
123	1111011
124	1111100
125	1111101
126	1111110
127	1111111
128	10000000
129	10000001
130	10000010
131	10000011
132	10000100
133	10000101
134	10000110
135	10000111
136	10001000
137	10001001
138	10001010
139	10001011
140	10001100
141	10001101
142	10001110
143	10001111
144	10010000
145	10010001
146	10010010
147	10010011
148	10010100
149	10010101
150	10010110
151	10010111
152	10011000
153	10011001
154	10011010
155	10011011
156	10011100
157	10011101
158	10011110
159	10011111
160	10100000
161	10100001
162	10100010

163	10100011
164	10100100
165	10100101
166	10100110
167	10100111
168	10101000
169	10101001
170	10101010
171	10101011
172	10101100
173	10101101
174	10101110
175	10101111
176	10110000
177	10110001
178	10110010
179	10110011
180	10110100
181	10110101
182	10110110
183	10110111
184	10111000
185	10111001
186	10111010
187	10111011
188	10111100
189	10111101
190	10111110
191	10111111
192	11000000
193	11000001
194	11000010
195	11000011
196	11000100
197	11000101
198	11000110
199	11000111
200	11001000
201	11001001
202	11001010
203	11001011
204	11001100
205	11001101
206	11001110
207	11001111
208	11010000
209	11010001
210	11010010
211	11010011
212	11010100
213	11010101
214	11010110
215	11010111
216	11011000
217	11011001
218	11011010
219	11011011
220	11011100
221	11011101
222	11011110
223	11011111
224	11100000
225	11100001
226	11100010

227	11100011
228	11100100
229	11100101
230	11100110
231	11100111
232	11101000
233	11101001
234	11101010
235	11101011
236	11101100
237	11101101
238	11101110
239	11101111
240	11110000
241	11110001
242	11110010
243	11110011
244	11110100
245	11110101
246	11110110
247	11110111
248	11111000
249	11111001
250	11111010
251	11111011
252	11111100
253	11111101
254	11111110
255	11111111
256	100000000
257	100000001
258	100000010
259	100000011
260	100000100
261	100000101
262	100000110
263	100000111
264	100001000
265	100001001
266	100001010
267	100001011
268	100001100
269	100001101
270	100001110
271	100001111
272	100010000
273	100010001
274	100010010
275	100010011
276	100010100
277	100010101
278	100010110
279	100010111
280	100011000
281	100011001
282	100011010
283	100011011
284	100011100
285	100011101
286	100011110
287	100011111
288	100100000
289	100100001
290	100100010

291	100100011
292	100100100
293	100100101
294	100100110
295	100100111
296	100101000
297	100101001
298	100101010
299	100101011
300	100101100
301	100101101
302	100101110
303	100101111
304	100110000
305	100110001
306	100110010
307	100110011
308	100110100
309	100110101
310	100110110
311	100110111
312	100111000
313	100111001
314	100111010
315	100111011
316	100111100
317	100111101
318	100111110
319	100111111
320	101000000
321	101000001
322	101000010
323	101000011
324	101000100
325	101000101
326	101000110
327	101000111
328	101001000
329	101001001
330	101001010
331	101001011
332	101001100
333	101001101
334	101001110
335	101001111
336	101010000
337	101010001
338	101010010
339	101010011
340	101010100
341	101010101
342	101010110
343	101010111
344	101011000
345	101011001
346	101011010
347	101011011
348	101011100
349	101011101
350	101011110
351	101011111
352	101100000
353	101100001
354	101100010

355	101100011
356	101100100
357	101100101
358	101100110
359	101100111
360	101101000
361	101101001
362	101101010
363	101101011
364	101101100
365	101101101
366	101101110
367	101101111
368	101110000
369	101110001
370	101110010
371	101110011
372	101110100
373	101110101
374	101110110
375	101110111
376	101111000
377	101111001
378	101111010
379	101111011
380	101111100
381	101111101
382	101111110
383	101111111
384	110000000
385	110000001
386	110000010
387	110000011
388	110000100
389	110000101
390	110000110
391	110000111
392	110001000
393	110001001
394	110001010
395	110001011
396	110001100
397	110001101
398	110001110
399	110001111
400	110010000
401	110010001
402	110010010
403	110010011
404	110010100
405	110010101
406	110010110
407	110010111
408	110011000
409	110011001
410	110011010
411	110011011
412	110011100
413	110011101
414	110011110
415	110011111
416	110100000
417	110100001
418	110100010

419	110100011
420	110100100
421	110100101
422	110100110
423	110100111
424	110101000
425	110101001
426	110101010
427	110101011
428	110101100
429	110101101
430	110101110
431	110101111
432	110110000
433	110110001
434	110110010
435	110110011
436	110110100
437	110110101
438	110110110
439	110110111
440	110111000
441	110111001
442	110111010
443	110111011
444	110111100
445	110111101
446	110111110
447	110111111
448	111000000
449	111000001
450	111000010
451	111000011
452	111000100
453	111000101
454	111000110
455	111000111
456	111001000
457	111001001
458	111001010
459	111001011
460	111001100
461	111001101
462	111001110
463	111001111
464	111010000
465	111010001
466	111010010
467	111010011
468	111010100
469	111010101
470	111010110
471	111010111
472	111011000
473	111011001
474	111011010
475	111011011
476	111011100
477	111011101
478	111011110
479	111011111
480	111100000
481	111100001
482	111100010

483	111100011
484	111100100
485	111100101
486	111100110
487	111100111
488	111101000
489	111101001
490	111101010
491	111101011
492	111101100
493	111101101
494	111101110
495	111101111
496	111110000
497	111110001
498	111110010
499	111110011
500	111110100
501	111110101
502	111110110
503	111110111
504	111111000
505	111111001
506	111111010
507	111111011
508	111111100
509	111111101
510	111111110
511	111111111
512	1000000000
513	1000000001
514	1000000010
515	1000000011
516	1000000100
517	1000000101
518	1000000110
519	1000000111
520	1000001000
521	1000001001
522	1000001010
523	1000001011
524	1000001100
525	1000001101
526	1000001110
527	1000001111
528	1000010000
529	1000010001
530	1000010010
531	1000010011
532	1000010100
533	1000010101
534	1000010110
535	1000010111
536	1000011000
537	1000011001
538	1000011010
539	1000011011
540	1000011100
541	1000011101
542	1000011110
543	1000011111
544	1000100000
545	1000100001
546	1000100010

547	1000100011
548	1000100100
549	1000100101
550	1000100110
551	1000100111
552	1000101000
553	1000101001
554	1000101010
555	1000101011
556	1000101100
557	1000101101
558	1000101110
559	1000101111
560	1000110000
561	1000110001
562	1000110010
563	1000110011
564	1000110100
565	1000110101
566	1000110110
567	1000110111
568	1000111000
569	1000111001
570	1000111010
571	1000111011
572	1000111100
573	1000111101
574	1000111110
575	1000111111
576	1001000000
577	1001000001
578	1001000010
579	1001000011
580	1001000100
581	1001000101
582	1001000110
583	1001000111
584	1001001000
585	1001001001
586	1001001010
587	1001001011
588	1001001100
589	1001001101
590	1001001110
591	1001001111
592	1001010000
593	1001010001
594	1001010010
595	1001010011
596	1001010100
597	1001010101
598	1001010110
599	1001010111
600	1001011000
601	1001011001
602	1001011010
603	1001011011
604	1001011100
605	1001011101
606	1001011110
607	1001011111
608	1001100000
609	1001100001
610	1001100010

611	1001100011
612	1001100100
613	1001100101
614	1001100110
615	1001100111
616	1001101000
617	1001101001
618	1001101010
619	1001101011
620	1001101100
621	1001101101
622	1001101110
623	1001101111
624	1001110000
625	1001110001
626	1001110010
627	1001110011
628	1001110100
629	1001110101
630	1001110110
631	1001110111
632	1001111000
633	1001111001
634	1001111010
635	1001111011
636	1001111100
637	1001111101
638	1001111110
639	1001111111
640	1010000000
641	1010000001
642	1010000010
643	1010000011
644	1010000100
645	1010000101
646	1010000110
647	1010000111
648	1010001000
649	1010001001
650	1010001010
651	1010001011
652	1010001100
653	1010001101
654	1010001110
655	1010001111
656	1010010000
657	1010010001
658	1010010010
659	1010010011
660	1010010100
661	1010010101
662	1010010110
663	1010010111
664	1010011000
665	1010011001
666	1010011010
667	1010011011
668	1010011100
669	1010011101
670	1010011110
671	1010011111
672	1010100000
673	1010100001
674	1010100010

675 1010100011
676 1010100100
677 1010100101
678 1010100110
679 1010100111
680 1010101000
681 1010101001
682 1010101010
683 1010101011
684 1010101100
685 1010101101
686 1010101110
687 1010101111
688 1010110000
689 1010110001
690 1010110010
691 1010110011
692 1010110100
693 1010110101
694 1010110110
695 1010110111
696 1010111000
697 1010111001
698 1010111010
699 1010111011
700 1010111100
701 1010111101
702 1010111110
703 1010111111
704 1011000000
705 1011000001
706 1011000010
707 1011000011
708 1011000100
709 1011000101
710 1011000110
711 1011000111
712 1011001000
713 1011001001
714 1011001010
715 1011001011
716 1011001100
717 1011001101
718 1011001110
719 1011001111
720 1011010000
721 1011010001
722 1011010010
723 1011010011
724 1011010100
725 1011010101
726 1011010110
727 1011010111
728 1011011000
729 1011011001
730 1011011010
731 1011011011
732 1011011100
733 1011011101
734 1011011110
735 1011011111
736 1011100000
737 1011100001
738 1011100010

739	1011100011
740	1011100100
741	1011100101
742	1011100110
743	1011100111
744	1011101000
745	1011101001
746	1011101010
747	1011101011
748	1011101100
749	1011101101
750	1011101110
751	1011101111
752	1011110000
753	1011110001
754	1011110010
755	1011110011
756	1011110100
757	1011110101
758	1011110110
759	1011110111
760	1011111000
761	1011111001
762	1011111010
763	1011111011
764	1011111100
765	1011111101
766	1011111110
767	1011111111
768	1100000000
769	1100000001
770	1100000010
771	1100000011
772	1100000100
773	1100000101
774	1100000110
775	1100000111
776	1100001000
777	1100001001
778	1100001010
779	1100001011
780	1100001100
781	1100001101
782	1100001110
783	1100001111
784	1100010000
785	1100010001
786	1100010010
787	1100010011
788	1100010100
789	1100010101
790	1100010110
791	1100010111
792	1100011000
793	1100011001
794	1100011010
795	1100011011
796	1100011100
797	1100011101
798	1100011110
799	1100011111
800	1100100000
801	1100100001
802	1100100010

803 1100100011
804 1100100100
805 1100100101
806 1100100110
807 1100100111
808 1100101000
809 1100101001
810 1100101010
811 1100101011
812 1100101100
813 1100101101
814 1100101110
815 1100101111
816 1100110000
817 1100110001
818 1100110010
819 1100110011
820 1100110100
821 1100110101
822 1100110110
823 1100110111
824 1100111000
825 1100111001
826 1100111010
827 1100111011
828 1100111100
829 1100111101
830 1100111110
831 1100111111
832 1101000000
833 1101000001
834 1101000010
835 1101000011
836 1101000100
837 1101000101
838 1101000110
839 1101000111
840 1101001000
841 1101001001
842 1101001010
843 1101001011
844 1101001100
845 1101001101
846 1101001110
847 1101001111
848 1101010000
849 1101010001
850 1101010010
851 1101010011
852 1101010100
853 1101010101
854 1101010110
855 1101010111
856 1101011000
857 1101011001
858 1101011010
859 1101011011
860 1101011100
861 1101011101
862 1101011110
863 1101011111
864 1101100000
865 1101100001
866 1101100010

867 1101100011
868 1101100100
869 1101100101
870 1101100110
871 1101100111
872 1101101000
873 1101101001
874 1101101010
875 1101101011
876 1101101100
877 1101101101
878 1101101110
879 1101101111
880 1101110000
881 1101110001
882 1101110010
883 1101110011
884 1101110100
885 1101110101
886 1101110110
887 1101110111
888 1101111000
889 1101111001
890 1101111010
891 1101111011
892 1101111100
893 1101111101
894 1101111110
895 1101111111
896 1110000000
897 1110000001
898 1110000010
899 1110000011
900 1110000100
901 1110000101
902 1110000110
903 1110000111
904 1110001000
905 1110001001
906 1110001010
907 1110001011
908 1110001100
909 1110001101
910 1110001110
911 1110001111
912 1110010000
913 1110010001
914 1110010010
915 1110010011
916 1110010100
917 1110010101
918 1110010110
919 1110010111
920 1110011000
921 1110011001
922 1110011010
923 1110011011
924 1110011100
925 1110011101
926 1110011110
927 1110011111
928 1110100000
929 1110100001
930 1110100010

931	1110100011
932	1110100100
933	1110100101
934	1110100110
935	1110100111
936	1110101000
937	1110101001
938	1110101010
939	1110101011
940	1110101100
941	1110101101
942	1110101110
943	1110101111
944	1110110000
945	1110110001
946	1110110010
947	1110110011
948	1110110100
949	1110110101
950	1110110110
951	1110110111
952	1110111000
953	1110111001
954	1110111010
955	1110111011
956	1110111100
957	1110111101
958	1110111110
959	1110111111
960	1111000000
961	1111000001
962	1111000010
963	1111000011
964	1111000100
965	1111000101
966	1111000110
967	1111000111
968	1111001000
969	1111001001
970	1111001010
971	1111001011
972	1111001100
973	1111001101
974	1111001110
975	1111001111
976	1111010000
977	1111010001
978	1111010010
979	1111010011
980	1111010100
981	1111010101
982	1111010110
983	1111010111
984	1111011000
985	1111011001
986	1111011010
987	1111011011
988	1111011100
989	1111011101
990	1111011110
991	1111011111
992	1111100000
993	1111100001
994	1111100010

```
995     1111100011
996     1111100100
997     1111100101
998     1111100110
999     1111100111
1000    1111101000
```

Write a function called root that is given a number x and an integer n and returns x1/n. In the function definition, set the default value of n to 2

In [14]:
```python
def root(x, n=2):
    return x ** (1/n)
result = root(16)
print(result)

result = root(8, 3)
print(result)

result = root(27, 3)
print(result)
```

```
4.0
2.0
3.0
```

An application of math.floor function is rounding a value to the nearest integer. Thestatement y = floor(x + .5) will round the number x to the nearest integer, and assign the result to y. Write a program that reads several numbers and uses the preceding statement to round each of these numbers to the nearest integer. For each number processed, print both the original number and the rounded number.

In [16]:
```python
import math

# Read numbers from the user and round them to the nearest integer
while True:
    input_str = input("Enter a number (or 'q' to quit): ")
    if input_str == 'q':
        break

    try:
        number = float(input_str)
        rounded_number = math.floor(number + 0.5)
        print("Original number:", number)
        print("Rounded number:", rounded_number)
    except ValueError:
        print("Invalid input. Please enter a valid number or 'q' to quit.")
```

```
Enter a number (or 'q' to quit): 343.5
Original number: 343.5
Rounded number: 344
Enter a number (or 'q' to quit): q
```

● Function floor may be used to round a number to a specific decimal place. The statement y = floor(x *10 + .5) / 10 rounds x to the tenths position (the first position to the right of the decimal point). Thestatement y = floor(x* 100 + .5) / 100 rounds x to the hundredths position (i.e., the second position to the right of the decimal point). Write a program that defines four functions to round a number x in various ways 7

a) roundToInteger(number)

b) roundToTenths(number)

c) roundToHundreths(number)

d) roundToThousandths(number)

For each value read, your program should print the original value, the number rounded to

the nearest integer, the number rounded to the nearest tenth, the number rounded to the

nearest hundredth, and the number rounded to the nearest thousandt

In [17]:
```python
import math

def roundToInteger(number):
    rounded_number = math.floor(number)
    return rounded_number

def roundToTenths(number):
    rounded_number = math.floor(number * 10 + 0.5) / 10
```

```
    return rounded_number

def roundToHundredths(number):
    rounded_number = math.floor(number * 100 + 0.5) / 100
    return rounded_number

def roundToThousandths(number):
    rounded_number = math.floor(number * 1000 + 0.5) / 1000
    return rounded_number

while True:
    input_str = input("Enter a number (or 'q' to quit): ")
    if input_str == 'q':
        break

    try:
        number = float(input_str)
        print("Original number:", number)
        print("Rounded to integer:", roundToInteger(number))
        print("Rounded to tenths:", roundToTenths(number))
        print("Rounded to hundredths:", roundToHundredths(number))
        print("Rounded to thousandths:", roundToThousandths(number))
    except ValueError:
        print("Invalid input. Please enter a valid number or 'q' to quit.")
```

```
Enter a number (or 'q' to quit): 1
Original number: 1.0
Rounded to integer: 1
Rounded to tenths: 1.0
Rounded to hundredths: 1.0
Rounded to thousandths: 1.0
Enter a number (or 'q' to quit): 4
Original number: 4.0
Rounded to integer: 4
Rounded to tenths: 4.0
Rounded to hundredths: 4.0
Rounded to thousandths: 4.0
Enter a number (or 'q' to quit): 3
Original number: 3.0
Rounded to integer: 3
Rounded to tenths: 3.0
Rounded to hundredths: 3.0
Rounded to thousandths: 3.0
Enter a number (or 'q' to quit): q
```

In [ ]: