Walter Cazzola

Home Page
ADAPT Lab.
Curriculum Vitae
Research Topic

Didactics

Publications

Funded Projects

Research Projects

Related Events

Exam of Advance in Programming

23 July 2012

Disclaimer. Note that to have a running solution for an exercise is not enough: you need a well-cooked solution that proves your ability to use what explained during the classes. The marks for the first 2 exercises is 12 each the marks for the one remaining is 8. To pass the exam you have to do, at least, two exercises. The submission with only one exercise will not be evaluated at all.

Exercise 1: Testing Sudoku Grids.

Sudoku is a logic-based, combinatorial number-placement puzzle. The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 sub-grids that compose the grid contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which typically has a unique solution.

The exercise consists of writing a function `is_sudoku` that given a filled grid checks if it is a valid sudoku, i.e., it respect **all** the rules of the game. The only admissible solution **must** be completely functional, i.e., for, while or similar control flow instructions are forbidden as well as the use of the itertools library.

The following is an example of the expected computation.

```
from issudoku import *

s0 = [[2,7,6,3,1,4,9,5,8],
      [8,5,4,9,6,2,7,1,3],
      [9,1,3,8,7,5,2,6,4],
      [4,6,8,1,2,7,3,9,5],
      [5,9,7,4,3,8,6,2,1],
      [1,3,2,5,9,6,4,8,7],
      [3,2,5,7,8,9,1,4,6],
      [6,4,1,2,5,3,8,7,9],
      [7,8,9,6,4,1,5,3,2]]

s1 = [[2,7,6,3,1,4,9,5,8],
      [8,5,4,9,6,2,7,1,3],
      [9,1,3,8,7,5,2,6,4],
      [4,6,8,1,2,7,3,9,5],
      [5,9,7,4,3,8,6,2,1],
      [3,2,5,7,8,9,1,4,6],
      [1,3,2,5,9,6,4,8,7],
      [6,4,1,2,5,3,8,7,9],
      [7,8,9,6,4,1,5,3,2]]

s2 = [[2,7,6,3,1,4,9,5,8],
      [8,5,4,9,6,2,7,1,3],
      [9,1,3,8,7,5,2,6,4],
      [3,2,5,7,8,9,1,4,6],
      [6,4,1,2,5,3,8,7,9],
      [7,8,9,6,4,1,5,3,2],
      [3,2,5,7,8,9,1,4,6],
      [6,4,1,2,5,3,8,7,9],
      [7,8,9,6,4,1,5,3,2]]

s3 = [[2,7,6,9,5,8,9,5,8],
      [8,5,4,7,1,3,7,1,3],
      [9,1,3,2,6,4,2,6,4],
      [4,6,8,3,9,5,3,9,5],
      [5,9,7,6,2,1,6,2,1],
      [1,3,2,4,8,7,4,8,7],
      [3,2,5,1,4,6,1,4,6],
      [6,4,1,8,7,9,8,7,9],
      [7,8,9,5,3,2,5,3,2]]

if __name__ == "__main__":
  print("s0 :-", is_sudoku(s0))
  print("s1 :-", is_sudoku(s1))
```

```
[17:11]cazzola@surtur:~/pa/es1>python3 main-issudoku.py
s0 :- True
s1 :- False
s2 :- False
```

Exercise 2: Detecting Getters and Setters Presence.

One of the main characteristics of object-oriented programming is the data hiding that permits to forbid the direct access to the state of an object. Such an approach protects a class developer from having to notify the users of his work that something will change and forces the users to update the state of the object in the correct way, i.e., by accessing the state through getters and setters.

In python, data hiding is just a convention but nothing prevents from writing our classes keeping this principle in mind and above all to our teachers to check if we are doing it well. Let's go and help the teachers.

In this exercise we have to use test driven development to write a test suite, named `SelectorPresenceTest` that automatically checks when is given a class if its attributes have a getter and setter method. To simplify the work the getter and setter methods **always** start for `get` and `set` respectively.

The following is an example of the expected behavior, note that, the unit test stops at the first missing selector (per kind) doesn't matter if there are many other missing selectors.

```python
from tddselector import *

class TestClass:
  def __init__(self, x, y):
    self.x = x
    self.y = y
    self.z = 0
  def getx(self): return x
  def getz(self): return z
  def setz(self, x): self.z = x

class TestClass2(TestClass):
  def sety(self, z): self.y=z
  def gety(self): return self.y

ClassesToTest = [
  SelectorPresenceTest(TestClass, TestClass(7,25)),
  SelectorPresenceTest(TestClass2, TestClass2(7,25))
]

if __name__ == "__main__":
  suite = unittest.TestSuite()
  for tc in ClassesToTest:
    suite.addTests(unittest.TestLoader().loadTestsFromTestCase(tc))
```

```
[17:33]cazzola@surtur:~/pa/es2>python3 main-tdd-selector.py
test_getters (__main__.SelectorPresenceTestInner) ... FAIL
test_setters (__main__.SelectorPresenceTestInner) ... FAIL
test_getters (__main__.SelectorPresenceTestInner) ... ok
test_setters (__main__.SelectorPresenceTestInner) ... FAIL

======================================================================
FAIL: test_getters (__main__.SelectorPresenceTestInner)
----------------------------------------------------------------------
Traceback (most recent call last):
                    [CUT]
AssertionError: Warning! gety doesn't exists!!!

======================================================================
FAIL: test_setters (__main__.SelectorPresenceTestInner)
----------------------------------------------------------------------
Traceback (most recent call last):
                    [CUT]
AssertionError: Warning! sety doesn't exists!!!

======================================================================
FAIL: test_setters (__main__.SelectorPresenceTestInner)
----------------------------------------------------------------------
Traceback (most recent call last):
                    [CUT]
AssertionError: Warning! setx doesn't exists!!!

----------------------------------------------------------------------
Ran 4 tests in 0.001s
```

Exercise 3: Gruenberger's Prime Path.

Among the prime numbers, 2 and 3 are exceptions to the rule that all primes are of one of two forms: **6k+1** or **6k-1**.

if you take a check sheet and draw a path of odd integers (one each check) starting from 3 that at every prime:

- turns left for primes of the form **6k+1**
- turns right for primes of the form **6k-1**

Eventually the path will cross itself, so that a check will contain two or more odd numbers. This exercise goes under the name of Gruenberger's prime path after the name of its creator.

Write a function `crossing()` that calculates the path intersections in the path from 3 to 10001 and lists them sorted by the smallest the largest table-wise a row for each list of intersections.

Note. The exercise must be solved in a functional manner any other solution will be considered wrong.

```python
from gruenberger import *

if __name__ == "__main__":
  for i in crossing():
```

```
### :-  [5, 1621]
### :-  [7, 1623]
### :-  [9, 4725]
### :-  [11, 1263]
### :-  [13, 1265]
### :-  [19, 1635]
### :-  [21, 1637]
### :-  [25, 7537]
### :-  [27, 7319, 7539]
### :-  [29, 7505, 7541]
### :-  [31, 1643, 7323, 7503, 7543]
### :-  [33, 1645, 7325]
### :-  [35, 1647, 7327]
### :-  [37, 1649, 7329]
### :-  [47, 1671]
### :-  [49, 1669]
### :-  [55, 1663]
### :-  [57, 1661]
### :-  [59, 1659]
### :-  [63, 1655]
### :-  [65, 7337]
### :-  [67, 4751]
```

```
### :-  [71, 7479]
### :-  [73, 7481]
### :-  [83, 7347, 7471]
### :-  [85, 7349, 7469]
### :-  [87, 7467]
### :-  [89, 7465]
### :-  [91, 7355]
### :-  [103, 4779]
### :-  [105, 4781]
### :-  [107, 4783]
### :-  [137, 337]
### :-  [145, 1357]
### :-  [147, 683]
### :-  [149, 349, 685]
### :-  [151, 1375]
### :-  [153, 1373]
### :-  [155, 355]
### :-  [159, 691]
### :-  [161, 693]
### :-  [163, 695]
### :-  [167, 1367]
### :-  [171, 319]
### :-  [175, 315]
### :-  [177, 701]
### :-  [179, 703]
### :-  [185, 769]
### :-  [187, 771]
### :-  [189, 773]
### :-  [245, 269]
### :-  [247, 271]
### :-  [255, 2351]
### :-  [257, 2349]
### :-  [263, 2339]
### :-  [281, 733]
### :-  [283, 735]
### :-  [289, 725]
### :-  [299, 715]
### :-  [301, 761]
### :-  [303, 759]
### :-  [305, 757]
### :-  [343, 1359]
### :-  [351, 687, 1371]
### :-  [353, 689]
### :-  [357, 2077]
### :-  [361, 673]
### :-  [363, 675]
### :-  [365, 677]
### :-  [367, 1351, 2067]
### :-  [369, 1349, 2065]
### :-  [371, 1347, 2063]
### :-  [373, 1345]
### :-  [379, 667]
### :-  [381, 669, 2057]
### :-  [383, 671]
### :-  [391, 631]
### :-  [393, 633]
### :-  [395, 635]
### :-  [397, 637]
### :-  [401, 593]
### :-  [403, 591]
### :-  [405, 589]
### :-  [407, 587]
### :-  [409, 2033]
### :-  [411, 2031]
### :-  [413, 2029]
### :-  [417, 2117, 9949]
### :-  [419, 575, 9947]
### :-  [421, 573, 2021]
### :-  [423, 819, 2019]
### :-  [425, 549, 2017, 9989]
### :-  [431, 539]
### :-  [433, 537]
### :-  [465, 2197]
### :-  [481, 2185]
### :-  [501, 2165]
### :-  [519, 2151]
### :-  [545, 9941]
### :-  [547, 9943, 9991]
### :-  [551, 2015, 9987]
```

```
### :-  [553, 2013, 9985]
### :-  [555, 2011, 9983]
### :-  [559, 1499]
### :-  [561, 1497]
### :-  [563, 831, 1495, 2127, 9959]
### :-  [565, 829, 2005, 2125, 9957]
### :-  [567, 2123, 9955]
### :-  [569, 2121, 9953]
### :-  [571, 823]
### :-  [579, 2115]
### :-  [605, 2109]
### :-  [607, 811]
### :-  [615, 2103]
### :-  [643, 2043]
### :-  [645, 2041]
### :-  [647, 2039]
### :-  [649, 1413]
### :-  [657, 1405]
### :-  [679, 2071]
### :-  [681, 1377]
### :-  [707, 767]
### :-  [783, 9899]
### :-  [805, 10001]
### :-  [807, 9999]
### :-  [809, 9997]
### :-  [817, 9945]
### :-  [827, 2007]
### :-  [833, 2129, 9961]
### :-  [835, 9963]
### :-  [837, 9965]
### :-  [839, 2139, 9151, 9967]
### :-  [841, 2137, 9153]
### :-  [843, 1483, 9155]
### :-  [845, 1481, 9157]
### :-  [847, 1439, 1527]
### :-  [849, 1441]
### :-  [851, 1443]
### :-  [853, 1445]
### :-  [857, 9169]
### :-  [859, 1451, 1555, 1959, 9171]
### :-  [861, 1557, 1957]
### :-  [863, 1559, 1955]
### :-  [865, 1457, 1561]
### :-  [867, 1563, 9203, 9239]
### :-  [869, 1565, 9201, 9237]
### :-  [871, 1567, 9199, 9235]
### :-  [873, 9233]
### :-  [875, 9231]
### :-  [877, 9229]
### :-  [879, 1075]
### :-  [881, 1577]
### :-  [883, 1579, 9223]
### :-  [885, 1581, 9221]
### :-  [887, 1583]
### :-  [889, 1585, 7601]
### :-  [891, 1587]
### :-  [893, 1589]
### :-  [895, 1591]
### :-  [897, 1593]
### :-  [899, 1095, 1595]
### :-  [901, 1597]
### :-  [903, 1139, 1247]
### :-  [905, 1105]
### :-  [909, 1609, 4673]
### :-  [911, 1607, 4671]
### :-  [915, 4647]
### :-  [917, 1713, 4713]
### :-  [923, 971, 1223]
### :-  [925, 969, 1221]
### :-  [927, 967, 1219, 1723]
### :-  [929, 989, 1217, 1725]
### :-  [931, 991]
### :-  [973, 1225, 1717, 4653, 4709]
### :-  [975, 1227]
### :-  [977, 1229]
### :-  [979, 1231]
### :-  [981, 4661]
### :-  [983, 4699]
### :-  [985, 1213, 4701]
### :-  [987, 1215, 4703]
```

```
### :-  [997, 1813]
### :-  [999, 1731, 1815]
### :-  [1001, 1817]
### :-  [1003, 1207, 1819]
### :-  [1005, 1821]
### :-  [1007, 1823]
### :-  [1009, 1161]
### :-  [1011, 1159]
### :-  [1013, 1157]
### :-  [1043, 7623]
### :-  [1065, 9213]
### :-  [1067, 1571]
### :-  [1069, 1569]
### :-  [1077, 9225]
### :-  [1079, 7595]
### :-  [1087, 1119]
### :-  [1089, 1117]
### :-  [1099, 1599, 4679]
### :-  [1101, 1141, 1245, 4677]
### :-  [1103, 4675]
### :-  [1107, 1251]
### :-  [1111, 1135]
### :-  [1137, 1249]
### :-  [1143, 1243, 1603]
### :-  [1145, 1241]
### :-  [1147, 1239]
### :-  [1149, 1237]
### :-  [1155, 4695]
### :-  [1165, 1825]
### :-  [1169, 1205]
### :-  [1175, 1199]
### :-  [1189, 1801]
### :-  [1191, 1751, 1803]
### :-  [1193, 1805]
### :-  [1195, 1755]
### :-  [1233, 4665]
### :-  [1257, 1617]
### :-  [1259, 4719]
### :-  [1261, 4721]
### :-  [1267, 1631, 4731]
### :-  [1271, 7551]
### :-  [1337, 1401]
### :-  [1353, 2069]
### :-  [1355, 1379]
### :-  [1421, 1973]
### :-  [1423, 1971]
### :-  [1431, 1983]
### :-  [1435, 1531]
### :-  [1437, 1529, 1989]
### :-  [1447, 1963]
### :-  [1449, 1553, 1961]
### :-  [1453, 9173]
### :-  [1459, 9243]
### :-  [1461, 1953, 9245]
### :-  [1463, 9247]
### :-  [1465, 9249]
### :-  [1467, 9251]
### :-  [1471, 9255]
### :-  [1477, 1521, 9161]
### :-  [1479, 9159]
### :-  [1485, 1993]
### :-  [1487, 1995]
### :-  [1489, 2133]
### :-  [1491, 1999, 2131]
### :-  [1493, 2001]
### :-  [1507, 9971]
### :-  [1509, 2145]
### :-  [1515, 9147]
### :-  [1523, 9163]
### :-  [1533, 1985]
### :-  [1539, 1967]
### :-  [1541, 9181]
### :-  [1543, 9183]
### :-  [1551, 9175]
### :-  [1601, 4681]
### :-  [1613, 4645]
### :-  [1615, 4643]
### :-  [1619, 4723]
### :-  [1629, 4729]
### :-  [1633, 4733]
```

```
### :-  [1641, 7321]
### :-  [1651, 7331]
### :-  [1675, 4599]
### :-  [1681, 4589, 7313]
### :-  [1683, 7531]
### :-  [1689, 7521]
### :-  [1693, 4625]
### :-  [1695, 4627]
### :-  [1697, 4629]
### :-  [1707, 4639]
### :-  [1709, 4641, 4717]
### :-  [1711, 4715]
### :-  [1715, 4651, 4711]
### :-  [1719, 4655, 4707]
### :-  [1721, 4657, 4705]
### :-  [1735, 1759]
### :-  [1737, 1761, 1809]
### :-  [1739, 1763]
### :-  [1741, 1765]
### :-  [1873, 9021]
### :-  [1875, 9023]
### :-  [1895, 1919]
### :-  [1899, 8455]
### :-  [1927, 7631]
### :-  [2023, 2119, 9951]
### :-  [2091, 9907]
### :-  [2095, 9911]
### :-  [2097, 9913]
### :-  [2099, 9915]
### :-  [2141, 9969]
### :-  [2237, 9801]
### :-  [2239, 9799]
### :-  [2251, 9787]
### :-  [2405, 2429]
### :-  [2607, 4987]
### :-  [2609, 4985]
### :-  [2615, 4995]
### :-  [2661, 2693]
### :-  [2663, 2691]
### :-  [2759, 5147]
### :-  [2761, 5149]
### :-  [2763, 5151]
### :-  [2765, 5153]
### :-  [2767, 5059]
### :-  [2785, 5041]
### :-  [2795, 5255]
### :-  [2805, 5265]
### :-  [2819, 4183]
### :-  [2821, 4181]
### :-  [2823, 4179]
### :-  [2825, 4177]
### :-  [2847, 5827]
### :-  [2849, 5829]
### :-  [2851, 5831]
### :-  [2857, 5817, 5973]
### :-  [2859, 5795, 5819, 5975]
### :-  [2861, 5821, 5977]
### :-  [2863, 6751]
### :-  [2867, 5991]
### :-  [2873, 2933]
### :-  [2885, 2921]
### :-  [2887, 5431]
### :-  [2889, 5429]
### :-  [2891, 2915, 5427]
### :-  [2893, 5425]
### :-  [2895, 5423]
### :-  [2897, 5421, 5457]
### :-  [2899, 3143, 5455]
### :-  [2901, 5453, 5613]
### :-  [2903, 5451]
### :-  [2909, 3133, 5441]
### :-  [2911, 3135, 5607]
### :-  [2913, 3137]
### :-  [2965, 3209]
### :-  [2967, 3207, 5475]
### :-  [2969, 3205]
### :-  [2983, 5419]
### :-  [2985, 3145, 5417]
### :-  [2987, 5615]
### :-  [2991, 5379, 5403]
```

```
### :-  [2993, 5513]
### :-  [2997, 5389]
### :-  [2999, 3875, 6567]
### :-  [3001, 3877, 6569]
### :-  [3003, 3879]
### :-  [3005, 3881]
### :-  [3019, 3111, 3895]
### :-  [3021, 3109, 3897]
### :-  [3023, 3899]
### :-  [3029, 6697]
### :-  [3031, 6119]
### :-  [3035, 6655]
### :-  [3039, 6047]
### :-  [3041, 6049, 6661]
### :-  [3043, 3079]
### :-  [3045, 3081]
### :-  [3047, 3083, 6091]
### :-  [3049, 6089]
### :-  [3051, 6087]
### :-  [3053, 6061, 6085]
### :-  [3055, 6083]
### :-  [3057, 6081]
### :-  [3059, 6079]
### :-  [3067, 3943]
### :-  [3075, 6055]
### :-  [3085, 6093]
### :-  [3087, 6095]
### :-  [3089, 6097, 6645]
### :-  [3091, 6039, 6647]
### :-  [3093, 6649]
### :-  [3095, 6651]
### :-  [3097, 6653]
### :-  [3101, 6117]
### :-  [3103, 6699]
### :-  [3113, 3893]
### :-  [3115, 3891]
### :-  [3117, 3889]
### :-  [3119, 6583]
### :-  [3121, 6581]
### :-  [3123, 6579]
### :-  [3125, 6577]
### :-  [3131, 5371, 5443, 5523]
### :-  [3147, 5415]
### :-  [3149, 5413]
### :-  [3151, 6547]
### :-  [3153, 5497, 5625, 6545]
### :-  [3155, 6543]
### :-  [3157, 6541]
### :-  [3159, 3847, 6539]
### :-  [3161, 3845, 6537]
### :-  [3163, 3843, 6535]
### :-  [3173, 3749]
### :-  [3175, 3715]
### :-  [3187, 3703]
### :-  [3189, 3705]
### :-  [3191, 3707]
### :-  [3197, 3765, 5485, 5637]
### :-  [3217, 3697, 3773, 5645]
### :-  [3219, 3695, 3775, 5647]
### :-  [3221, 3693, 3777]
### :-  [3231, 3683, 3787]
### :-  [3245, 3621]
### :-  [3247, 3651]
### :-  [3257, 6929]
### :-  [3259, 6927]
### :-  [3281, 3345]
### :-  [3293, 3329, 6829]
### :-  [3295, 3327, 6827]
### :-  [3297, 3325]
### :-  [3299, 3323]
### :-  [3309, 6997]
### :-  [3311, 6995]
### :-  [3313, 6813, 6993]
### :-  [3315, 6815]
### :-  [3317, 6817]
### :-  [3319, 6819]
### :-  [3347, 6883]
### :-  [3349, 6885]
### :-  [3351, 6887]
### :-  [3353, 6889]
```

```
### :-  [3355, 6891]
### :-  [3357, 6893]
### :-  [3359, 6895]
### :-  [3365, 6901]
### :-  [3465, 3585]
### :-  [3489, 3533]
### :-  [3491, 3531]
### :-  [3495, 3735]
### :-  [3497, 3829]
### :-  [3501, 3729]
### :-  [3503, 3835]
### :-  [3511, 6519]
### :-  [3513, 6517]
### :-  [3515, 6515]
### :-  [3517, 6513]
### :-  [3591, 3803]
### :-  [3593, 3801]
### :-  [3615, 3655]
### :-  [3625, 3649]
### :-  [3671, 3799]
### :-  [3673, 3797]
### :-  [3675, 3795]
### :-  [3677, 3793]
### :-  [3679, 3791]
### :-  [3681, 3789]
### :-  [3685, 3785]
### :-  [3687, 3783]
### :-  [3689, 3781]
### :-  [3691, 3779]
### :-  [3721, 3745]
### :-  [3731, 3831]
### :-  [3741, 3825]
### :-  [3761, 5489, 5633]
### :-  [3763, 5487, 5635]
### :-  [3767, 5483, 5639]
### :-  [3769, 5641]
### :-  [3771, 5643]
### :-  [3837, 6529]
### :-  [3839, 6531]
### :-  [3841, 6533]
### :-  [3871, 6563]
### :-  [3873, 6565]
### :-  [3901, 6129]
### :-  [3935, 6679]
### :-  [3937, 6677]
### :-  [3939, 6675]
### :-  [3941, 6673]
### :-  [4061, 5221]
### :-  [4087, 5231]
### :-  [4089, 5229]
### :-  [4091, 5227]
### :-  [4093, 5225]
### :-  [4111, 5319]
### :-  [4113, 5317]
### :-  [4115, 5291, 5315]
### :-  [4117, 5313]
### :-  [4119, 5311]
### :-  [4121, 5309]
### :-  [4125, 5245]
### :-  [4145, 5253]
### :-  [4163, 5271]
### :-  [4231, 4967]
### :-  [4233, 4969]
### :-  [4281, 4833]
### :-  [4291, 4839]
### :-  [4311, 4907]
### :-  [4379, 4415]
### :-  [4433, 7393]
### :-  [4435, 7391]
### :-  [4437, 7389]
### :-  [4439, 7387]
### :-  [4441, 7385]
### :-  [4475, 7383]
### :-  [4477, 7437]
### :-  [4495, 7419]
### :-  [4501, 7409]
### :-  [4541, 7301]
### :-  [4585, 4609, 7309]
### :-  [4587, 7311]
```

```
### :-  [4591, 7315]
### :-  [4607, 7307]
### :-  [4737, 7549]
### :-  [4739, 7499, 7547]
### :-  [4741, 7497]
### :-  [4743, 7495]
### :-  [4745, 7493]
### :-  [4747, 7491]
### :-  [4749, 7489]
### :-  [4753, 7485]
### :-  [4793, 7365, 7453]
### :-  [4795, 7367, 7451]
### :-  [4797, 7369]
### :-  [5073, 5169]
### :-  [5079, 5175]
### :-  [5085, 5181]
### :-  [5099, 5123]
### :-  [5101, 5125]
### :-  [5321, 5933]
### :-  [5329, 5941]
### :-  [5337, 5557, 6621]
### :-  [5339, 5555]
### :-  [5341, 5553, 5853]
### :-  [5343, 5551]
### :-  [5345, 5549, 5957]
### :-  [5347, 5547]
### :-  [5349, 6609]
### :-  [5355, 5587, 6007, 6731]
### :-  [5357, 6009]
### :-  [5359, 6011]
### :-  [5367, 5527]
### :-  [5369, 5525]
### :-  [5373, 5445, 5521]
### :-  [5375, 5399, 5447]
### :-  [5377, 5401, 5449]
### :-  [5381, 5405]
### :-  [5383, 5511]
### :-  [5391, 6571]
### :-  [5393, 6573]
### :-  [5397, 5517]
### :-  [5407, 6555]
### :-  [5409, 6553]
### :-  [5411, 5619]
### :-  [5435, 5603]
### :-  [5491, 5631]
### :-  [5493, 5629]
### :-  [5495, 5627]
### :-  [5499, 5623]
### :-  [5503, 6551]
### :-  [5509, 6557]
### :-  [5539, 6015]
### :-  [5543, 5583, 6727]
### :-  [5545, 6605]
### :-  [5559, 6623]
### :-  [5561, 5945, 6625]
### :-  [5563, 6627]
### :-  [5571, 5859, 5951]
### :-  [5575, 5955]
### :-  [5579, 6603]
### :-  [5581, 6725]
### :-  [5585, 6729]
### :-  [5589, 6005, 6733]
### :-  [5591, 6003]
### :-  [5621, 6549]
### :-  [5715, 6767]
### :-  [5789, 5825]
### :-  [5797, 6749]
### :-  [5801, 5993]
### :-  [5803, 5995]
### :-  [5805, 5997]
### :-  [5807, 5999]
### :-  [5811, 6743]
### :-  [5813, 5969]
### :-  [5815, 5971]
### :-  [5849, 6613]
### :-  [5851, 6615]
### :-  [5857, 5949]
### :-  [5861, 5953]
### :-  [5875, 6643]
### :-  [5887, 6063]
```

```
### :-  [5893, 6077]
### :-  [5959, 6611]
### :-  [5983, 6755]
### :-  [6021, 6593]
### :-  [6023, 6715]
### :-  [6027, 6111]
### :-  [6051, 6663]
### :-  [6053, 6665]
### :-  [6123, 6695]
### :-  [6305, 6329]
### :-  [6307, 6331]
### :-  [6309, 6333]
### :-  [6311, 6335]
### :-  [6341, 6401]
### :-  [6347, 6395]
### :-  [6359, 6383, 6459]
### :-  [6361, 6385, 6457]
### :-  [6379, 6463]
### :-  [6381, 6461]
### :-  [6387, 6455]
### :-  [6389, 6453]
### :-  [6597, 6721]
### :-  [6833, 7037]
### :-  [6835, 7039]
### :-  [6845, 7049]
### :-  [6857, 7077]
### :-  [6859, 7075]
### :-  [6861, 7073]
### :-  [6863, 7071]
### :-  [6877, 7053]
### :-  [7091, 7127]
### :-  [7093, 7129]
### :-  [7317, 7509]
### :-  [7341, 7477]
### :-  [7343, 7475]
### :-  [7345, 7473]
### :-  [7361, 7457]
### :-  [7363, 7455]
### :-  [7501, 7545]
### :-  [7511, 7535]
### :-  [7695, 7779]
### :-  [7807, 7963]
### :-  [7809, 7961]
### :-  [7811, 7835, 7959]
### :-  [7813, 7957]
### :-  [7815, 7955]
### :-  [7817, 7953]
### :-  [7861, 7945]
### :-  [7881, 7937]
### :-  [7883, 7935]
### :-  [7885, 8073]
### :-  [7889, 7925]
### :-  [7895, 8099]
### :-  [7899, 8131]
### :-  [7909, 8125]
### :-  [7913, 8057]
### :-  [7921, 8065]
### :-  [7931, 8075]
### :-  [8105, 8129]
### :-  [8211, 8143]
### :-  [8265, 7901]
### :-  [8267, 7903, 8107]
### :-  [8269, 7905]
### :-  [8273, 8113]
### :-  [8275, 8115]
### :-  [8277, 8117]
### :-  [8299, 8335]
### :-  [8385, 8025]
### :-  [8409, 7985]
### :-  [8435, 9339]
### :-  [8437, 9337]
### :-  [8465, 1925]
### :-  [8473, 1893]
### :-  [8479, 9035]
### :-  [8531, 8995]
### :-  [8585, 8909]
### :-  [8663, 8619]
### :-  [8665, 8617]
### :-  [8667, 8615]
### :-  [8669, 8613]
```

```
### :-  [8679, 8767]
### :-  [8701, 8725]
### :-  [8731, 8691]
### :-  [8733, 8689]
### :-  [8751, 8647]
### :-  [8753, 8649]
### :-  [8775, 8611]
### :-  [8801, 8901]
### :-  [8811, 8883]
### :-  [8857, 8945]
### :-  [8941, 8561]
### :-  [8943, 8563]
### :-  [8959, 8843]
### :-  [9003, 8515]
### :-  [9025, 1877]
### :-  [9031, 1883]
### :-  [9037, 1913]
### :-  [9039, 1911]
### :-  [9041, 1909, 8449]
### :-  [9043, 8447]
### :-  [9045, 8445]
### :-  [9047, 8443]
### :-  [9063, 7659]
### :-  [9075, 9279]
### :-  [9093, 9261]
### :-  [9121, 9085]
### :-  [9205, 9241]
### :-  [9253, 1469]
### :-  [9267, 9119]
### :-  [9291, 7759]
### :-  [9293, 7757]
### :-  [9313, 7777]
### :-  [9327, 7679]
```

Last Modified: Fri, 03 Aug 2012 12:08:32 ADAPT Lab.