

```
In [1]: import time
import random
class PhoneCounter:
    def __init__(self):
        self.phones = {}

    def hit(self, number, timestamp):
        if number not in self.phones.keys():
            self.phones[number] = [timestamp]
        else:
            self.phones[number].append(timestamp)

    def getHits(self, timestamp, selector = None, duration = 300):
        if len(self.phones.keys()) == 0:
            return 0

        if selector == None:
            hits = 0
            for key, val in self.phones.items():
                ptr = len(val)-1
                while ptr >= 0 and abs(timestamp-val[ptr]) <= duration:
                    hits += 1
                    ptr -= 1
            return hits
        elif selector in self.phones.keys():
            hits = 0
            for val in self.phones[selector]:
                if abs(timestamp-val) <= duration:
                    hits += 1
            return hits

    def getSelector(self, number, timestamp = None, duration = 300):
        number = str(number)
        if number in self.phones.keys():
            try:
                if abs(timestamp-self.phones[number][-1]) <= duration:
                    return [x for x in self.phones[number] if abs(timestamp-x) <= duration]
                else:
                    return self.phones[number]
            except:
                return "Error"
        return "No selector here!"

    def database(self, number = None):
        if number == None:
            return self.phones.items()
        elif number in self.phones.keys():
            return self.phones[number]
        else:
            return "Number not found!"

In [2]: print(time.time()+100)

1615835680.0723925
```

```
In [3]: def generatorPhones(amount):
        arr = []
        for v in range(1, amount+1):
```

```
number = ""
for y in range(10):
    number += str(random.randint(0,9))
arr.append(number if int(number) != 0 else "1111111111")
return arr
```

```
In [4]: phones = generatorPhones(100)
```

```
In [5]: phoneCollector = PhoneCounter()
```

```
In [6]: for i in range(300):
         phoneCollector.hit(phones[i*100], time.time()+10*i)
```

```
In [7]: print(phoneCollector.database())
```

dict_items({'('834736433',), (1615835580.4627514, 1615836580.4627514, 1615837580.4627514), ('1038202715',), (1615835590.4627514, 1615836590.4627514, 1615837590.4627514), ('7180145843',), (1615835600.4627514, 1615836600.4627514, 1615837600.4627514), ('3166397889',), (1615835610.4627514, 1615836610.4627514, 1615837610.4627514), ('1924848541',), (1615835620.4627514, 1615836620.4627514, 1615837620.4627514), ('8118023850',), (1615835630.4627514, 1615836630.4627514, 1615837630.4627514), ('6274384689',), (1615835640.4627514, 1615836640.4627514, 1615837640.4627514), ('5907548321',), (1615835650.4627514, 1615836650.4627514, 1615836650.4627514, 1615837650.4627514), ('9634582318',), (1615835660.4627514, 1615836660.4627514, 1615837660.4627514), ('3678725381',), (1615835670.4627514, 1615836670.4627514, 1615837670.4627514), ('294091351',), (1615835680.4627514, 1615836680.4627514, 1615837680.4627514), ('4033390847',), (1615835690.4627514, 1615836690.4627514, 1615837690.4627514), ('3368220685',), (1615835700.4627514, 1615836700.4627514, 1615837700.4627514), ('3668983513',), (1615835710.4627514, 1615836710.4627514, 1615837710.4627514), ('9465128243',), (1615835720.4627514, 1615836720.4627514, 1615837720.4627514), ('3825872148',), (1615835730.4627514, 1615836730.4627514, 1615837730.4627514), ('5635816541',), (1615835740.4627514, 1615836740.4627514, 1615837740.4627514), ('0515001618',), (1615835750.4627514, 1615836750.4627514, 1615837750.4627514), ('6166834809',), (1615835760.4627514, 1615836760.4627514, 1615837760.4627514), ('80785123',), (1615835770.4627514, 1615836770.4627514, 1615837770.4627514), ('7548472324',), (1615835780.4627514, 1615836780.4627514, 1615837780.4627514), ('0655984094',), (1615835790.4627514, 1615836790.4627514, 1615837790.4627514), ('8400958160',), (1615835800.4627514, 1615836800.4627514, 1615837800.4627514), ('9937913827',), (1615835810.4627514, 1615836810.4627514, 1615837810.4627514), ('9369563353',), (1615835820.4627514, 1615836820.4627514, 1615837820.4627514), ('8863173158',), (1615835830.4627514, 1615836830.4627514, 1615837830.4627514), ('6158377035',), (1615835840.4627514, 1615836840.4627514, 1615837840.4627514), ('4894660672',), (1615835850.4627514, 1615836850.4627514, 1615837850.4627514), ('374383645',), (1615835860.4627514, 1615836860.4627514, 1615837860.4627514), ('2188259319',), (1615835870.4627514, 1615836870.4627514, 1615837870.4627514), ('5415647028',), (1615835880.4627514, 1615836880.4627514, 1615837880.4627514), ('9460086471',), (1615835890.4627514, 1615836890.4627514, 1615837890.4627514), ('8413994319',), (1615835900.4627514, 1615836900.4627514, 1615837900.4627514), ('6296716927',), (1615835910.4627514, 1615836910.4627514, 1615837910.4627514), ('4153936406',), (1615835920.4627514, 1615836920.4627514, 1615837920.4627514, 1615837930.4627514), ('5219828877',), (1615835930.4627514, 1615836930.4627514, 1615837930.4627514), ('6022515746',), (1615835940.4627514, 1615836940.4627514, 1615837940.4627514), ('125386229',), (1615835950.4627514, 1615836950.4627514, 1615837950.4627514), ('9487901495',), (1615835960.4627514, 1615836960.4627514, 1615837960.4627514), ('7782483244',), (1615835970.4627514, 1615836970.4627514, 1615837970.4627514), ('7209912441',), (1615835980.4627514, 1615836980.4627514, 1615837980.4627514), ('6743520733',), (1615835990.4627514, 1615836990.4627514, 1615837990.4627514), ('5594059846',), (1615836000.4627514, 1615837000.4627514, 1615838000.4627514), ('0352512954',), (1615836010.4627514, 1615837010.4627514, 1615838010.4627514, 1615838020.4627514), ('2524295513',), (1615836030.4627514, 1615837030.4627514, 1615838030.4627514), ('563502062',), (1615836040.4627514, 1615837040.4627514, 1615838040.4627514), ('8319551334',), (1615836050.4627514, 1615837050.4627514, 1615838050.4627514), ('0370951997',), (1615836060.4627514, 1615837060.4627514, 1615838060.4627514), ('5195143590',), (1615836070.4627514, 1615837070.4627514, 1615838070.4627514), ('7760980801',), (1615836080.4627514, 1615837080.4627514, 1615838080.4627514), ('1866172901',), (1615836090.4627514, 1615837090.4627514, 1615838090.4627514), ('2300967933',), (1615836100.4627514, 1615837100.4627514, 1615838100.4627514, 1615838110.4627514), ('1872759197',), (1615836120.4627514, 1615837120.4627514, 1615838120.4627514), ('709543102',), (1615836130.4627514, 1615837130.4627514, 1615838130.4627514), ('5809893541',), (1615836140.4627514, 1615837140.4627514, 1615838140.4627514), ('102

```
In [8]: print(phones)
for i in range(100):
    phoneCollector.getSelector(phones[i])

['83747386433', '1038204215', '7180145843', '3166397889', '1924848541', '8118023850', '6274384689', '5
907548321', '9634582318', '3678725381', '2940913151', '4033390847', '3368220685', '3668983513', '9465
218243', '3825872148', '5635816541', '0515001618', '6166834809', '8078551234', '7548472324', '0655984
094', '8400958160', '9937913827', '9369563353', '8863173158', '6753777035', '4894660672', '374383645
8', '2188259319', '5415647028', '9460086471', '8413994319', '6296716927', '4153936406', '5219882877',
'6022515746', '1253862296', '9487901495', '7782483244', '7209912441', '67433520733', '5594059846', '03
52512954', '3617298132', '2524295513', '5635020623', '8319525134', '0370951997', '5195143590', '77606
90801', '1866172901', '2300967933', '2519264429', '1872759197', '7095431024', '5808993541', '10256042
03', '0134595307', '9197909910', '7407592522', '8541806759', '6388562415', '0902073915', '362372980
7', '2196741278', '8261344298', '3420320588', '2441271194', '0577671076', '0590274407', '0685848095',
'1338462686', '0141802377', '6951277727', '3139027491', '9610520109', '8672480453', '087527833', '95
65146715', '3178805329', '9489316635', '9583602459', '9728898265', '0922802227', '2409454837', '88512
69016', '0985485181', '3063690515', '7037706695', '7484008598', '1052662995', '9275916450', '38228815
24', '4908503340', '5612164194', '8215033403', '6907970424', '7979501665', '5114376038']

In [9]: for i in range(100):
        print(phoneCollector.getHits(time.time()*100*i, duration = 2500))#, duration = 5)
```

```
In [10]: print(phoneCollector.getHits(time.time(), selector = phones[10], duration = 1000))
```

2940913151

```
In [12]: print(phoneCollector.database(number = phones[10]))
```

[1615835680.4627514, 1615836680.4627514, 1615837680.4627514]