```
import pandas as pd
celebs = pd.read_csv("./data/famousbday.txt",sep=':',header=None,
names=['name','occup','bday'])
celeb_mbti = pd.read_csv("./data/myer-briggs.txt",header=None,sep=':',\
names=['mbti','name'])
df = pd.merge(celeb_mbti,celebs)
print df[:4]
print df.shape
```

```
   mbti            name                   occup        bday
0  ENFJ  Abraham Maslow  American psychologist    1/4/1908
1  ENFJ     Ben Stiller         American actor  30/11/1965
2  ENFJ       Bob Saget         American actor   17/5/1956
3  ENFJ  Brenda Vaccaro       American actress  18/11/1939
(539, 4)
```

```
from datetime import datetime
def f(s):
   try:
      return datetime.strptime(s, '%d/%m/%Y').date().strftime('%Y%m%d')
   except:
      return None
print f('18/11/1939')
print f('18/11/1839')
df['bday'] = df['bday'].apply(f)
print df[:4]
```

```
19391118
None
   mbti            name                   occup      bday
0  ENFJ  Abraham Maslow  American psychologist  19080401
1  ENFJ     Ben Stiller         American actor  19651130
2  ENFJ       Bob Saget         American actor  19560517
3  ENFJ  Brenda Vaccaro       American actress  19391118
```

```
import mindmeld; reload(mindmeld)
import pprint
pprint.pprint (mindmeld.calculate('19080401'))
```

```
{'chinese': 'Rooster',
 'lewi': [1, 149, 155, 181, 182, 193, 194, 200, 230, 244],
 'millman': [2, 3, 5, '235'],
 'spiller': 'Cancer'}
```

```
import os
cols = []
lewi = os.listdir('./doc/details/lewi')
lewi = map(lambda x: x.replace('.html',''),lewi)
cols += lewi
millman = os.listdir('./doc/details/millman')
millman = map(lambda x: 'mill'+x.replace('.html',''),millman)
millman.remove('millnineyearcycle')
cols += millman
chinese = os.listdir('./doc/details/chinese')
chinese = map(lambda x: x.replace('.html',''),chinese)
```

```
cols += chinese
spiller = os.listdir('./doc/details/spiller')
spiller = map(lambda x: x.replace('.html',''),spiller)
cols += spiller
print np.array(cols)
```

```
['143' '53' '154' '64' '160' '55' '91' '140' '242' '155' '236' '206' '47'
 '275' '259' '111' '40' '166' '192' '174' '209' '51' '266' '10' '94' '269'
 '258' '36' '146' '7' '98' '151' '67' '48' '262' '71' '12' '219' '43' '2'
 '129' '96' '69' '114' '79' '28' '189' '229' '119' '196' '85' '131' '148'
 '124' '100' '19' '226' '244' '159' '52' '216' '185' '46' '257' '122' '125'
 '197' '271' '21' '249' '77' '215' '44' '106' '274' '32' '147' '223' '116'
 '3' '41' '56' '150' '88' '253' '101' '183' '123' '14' '35' '205' '130'
 '31' '60' '108' '54' '34' '92' '63' '195' '17' '115' '82' '135' '177'
 '120' '33' '25' '231' '153' '78' '169' '276' '188' '13' '186' '137' '212'
 '16' '24' '102' '152' '217' '20' '29' '277' '228' '139' '9' '30' '144'
 '156' '68' '161' '202' '250' '127' '198' '22' '81' '181' '194' '141' '128'
 '18' '233' '145' '168' '241' '132' '247' '158' '248' '65' '225' '232' '84'
 '182' '240' '1' '23' '49' '207' '37' '210' '273' '221' '8' '75' '113' '15'
 '76' '272' '199' '4' '224' '243' '11' '261' '134' '254' '104' '72' '103'
 '178' '245' '57' '26' '200' '136' '80' '237' '267' '227' '95' '184' '110'
 '117' '211' '190' '58' '126' '201' '93' '235' '180' '90' '255' '99' '107'
 '109' '97' '39' '66' '157' '170' '45' '27' '133' '171' '268' '238' '230'
 '6' '213' '175' '5' '234' '187' '263' '214' '70' '179' '260' '42' 'mod.pl'
 '251' '222' '121' '73' '86' '61' '191' '220' '62' '162' '138' '83' '270'
 '204' '167' '172' '59' '74' '218' '50' '87' '118' '208' '112' '164' '38'
 '239' '264' '203' '105' '165' '265' '193' '142' '256' '149' '173' '252'
 '89' '246' '176' '163' 'mill325' 'mill2911' 'mill7' 'mill2' 'mill189'
 'mill4610' 'mill437' 'mill257' 'mill4711' 'mill3' 'mill1910' 'mill123'
 'mill336' 'mill404' 'mill3912' 'mill4812' 'mill3811' 'mill426' 'mill9'
 'mill314' 'mill156' 'mill202' 'mill448' 'mill145' 'mill3710' 'mill1'
 'mill347' 'mill8' 'mill4' 'mill224' 'mill134' 'mill178' 'mill303' 'mill0'
 'mill235' 'mill459' 'mill268' 'mill6' 'mill213' 'mill5' 'mill358'
 'mill2810' 'mill369' 'mill279' 'mill415' 'mill167' 'mill246' 'Monkey'
 'Sheep' 'Rooster' 'Snake' 'Dragon' 'Dog' 'Rat' 'Pig' 'Tiger' 'Ox' 'Rabbit'
 'Horse' 'Libra' 'Scorpio' 'Aries' 'Cancer' 'Pisces' 'Gemini' 'Virgo'
 'Aquarius' 'Sagittarius' 'Taurus' 'Leo' 'Capricorn']
```

```
for x in cols: df[x] = np.nan
print df.ix[0]
```

```
mbti                        ENFJ
name              Abraham Maslow
occup    American psychologist
bday                    19080401
143                          NaN
53                           NaN
154                          NaN
64                           NaN
160                          NaN
55                           NaN
91                           NaN
140                          NaN
242                          NaN
155                          NaN
236                          NaN
```

```
...
Ox              NaN
Rabbit          NaN
Horse           NaN
Libra           NaN
Scorpio         NaN
Aries           NaN
Cancer          NaN
Pisces          NaN
Gemini          NaN
Virgo           NaN
Aquarius        NaN
Sagittarius     NaN
Taurus          NaN
Leo             NaN
Capricorn       NaN
Name: 0, Length: 353, dtype: object
```

```
df = df[pd.isnull(df['bday']) == False]
def enrich(x):
   res = mindmeld.calculate(x['bday'])
   for lew in res['lewi']: x[lew] = 1

print df.shape
df2 = df[:4].apply(enrich, axis=1)
df.to_csv('/tmp/out.csv',sep=';')
df2.to_csv('/tmp/out2.csv',sep=';')

(461, 353)
```