In [1]:
```
import csv
import pandas as pd
import numpy as np
```

In [2]:
```
from pandas.core.frame import DataFrame
```

In [3]:
```
# load csv file

datacsv = pd.read_csv('/Users/kloe/Desktop/Middle Office Analyst Test data file.csv',index_col=False)
```

In [4]:
```
print(datacsv)
```

```
      tx_id            time exchange currency    tx_type         Amoun
t
0     94719   2/25/20 7:23        B      BTC    deposit  6.208000e+0
5
1     47648   2/25/20 1:05        C      BCH    deposit  1.149200e+0
6
2     64351   2/25/20 6:09        D      EOS    deposit  2.309000e+0
5
3      1720   2/25/20 1:27        B      ETH    deposit  1.229187e+0
6
4     29500   2/25/20 3:35        C      BCH  withdrawl  9.873000e+0
5
...     ...            ...      ...      ...        ...           ..
.
2685  94252  2/25/20 13:20        B      USD  withdrawl  1.424700e+0
6
2686  14922   2/25/20 1:05        D      EOS  withdrawl  2.949426e+0
5
2687  59020  2/25/20 11:28        C      BTC    deposit  7.347316e+0
5
2688  74925   2/25/20 4:27        C      EOS    deposit  1.000000e+0
5
2689  24909  2/25/20 12:37        A      ETH    deposit  2.000000e+0
5

[2690 rows x 6 columns]
```

In [5]:
```
df = pd.DataFrame(datacsv)
print(df)
```

```
      tx_id            time exchange currency    tx_type        Amoun
t
0     94719   2/25/20 7:23        B      BTC    deposit  6.208000e+0
5
1     47648   2/25/20 1:05        C      BCH    deposit  1.149200e+0
6
2     64351   2/25/20 6:09        D      EOS    deposit  2.309000e+0
5
3      1720   2/25/20 1:27        B      ETH    deposit  1.229187e+0
6
4     29500   2/25/20 3:35        C      BCH  withdrawl  9.873000e+0
5
...     ...            ...      ...      ...        ...           ..
.
2685  94252  2/25/20 13:20        B      USD  withdrawl  1.424700e+0
6
2686  14922   2/25/20 1:05        D      EOS  withdrawl  2.949426e+0
5
2687  59020  2/25/20 11:28        C      BTC    deposit  7.347316e+0
5
2688  74925   2/25/20 4:27        C      EOS    deposit  1.000000e+0
5
2689  24909  2/25/20 12:37        A      ETH    deposit  2.000000e+0
5

[2690 rows x 6 columns]
```

In [6]:
```
# add an empty column for 'Flag any transfer that started before 8am a
nd finished after 8am.'

df['flag'] = [[] for _ in range(len(df))]
```

In [7]:
```
print(df.dtypes)
```

```
tx_id         int64
time         object
exchange     object
currency     object
tx_type      object
Amount      float64
flag         object
dtype: object
```

In [8]:
```
print(df)
```

```
      tx_id           time exchange currency    tx_type        Amoun
t flag
0     94719   2/25/20 7:23        B      BTC    deposit  6.208000e+0
5   []
1     47648   2/25/20 1:05        C      BCH    deposit  1.149200e+0
6   []
2     64351   2/25/20 6:09        D      EOS    deposit  2.309000e+0
5   []
3      1720   2/25/20 1:27        B      ETH    deposit  1.229187e+0
6   []
4     29500   2/25/20 3:35        C      BCH  withdrawl  9.873000e+0
5   []
...     ...            ...      ...      ...        ...           ..
.  ...
2685  94252  2/25/20 13:20        B      USD  withdrawl  1.424700e+0
6   []
2686  14922   2/25/20 1:05        D      EOS  withdrawl  2.949426e+0
5   []
2687  59020  2/25/20 11:28        C      BTC    deposit  7.347316e+0
5   []
2688  74925   2/25/20 4:27        C      EOS    deposit  1.000000e+0
5   []
2689  24909  2/25/20 12:37        A      ETH    deposit  2.000000e+0
5   []

[2690 rows x 7 columns]
```

In [9]:
```
# transfer object type to datatime type

df['time'] = pd.to_datetime(df['time'])
```

In [10]:
```
print(df.dtypes)
```

```
tx_id                 int64
time         datetime64[ns]
exchange             object
currency             object
tx_type              object
Amount              float64
flag                 object
dtype: object
```

In [11]:
```
# Flag any transfer that started before 8am

df['flag'] = np.where(df['time']>'2/25/20 8:00','started b4 8am','')
```

In [12]:
```
print(df)
```

```
      tx_id                time exchange currency    tx_type
Amount  \
0     94719 2020-02-25 07:23:00        B      BTC    deposit  6.2080
00e+05
1     47648 2020-02-25 01:05:00        C      BCH    deposit  1.1492
00e+06
2     64351 2020-02-25 06:09:00        D      EOS    deposit  2.3090
00e+05
3      1720 2020-02-25 01:27:00        B      ETH    deposit  1.2291
87e+06
4     29500 2020-02-25 03:35:00        C      BCH  withdrawl  9.8730
00e+05
...     ...                 ...      ...      ...        ...
...
2685  94252 2020-02-25 13:20:00        B      USD  withdrawl  1.4247
00e+06
2686  14922 2020-02-25 01:05:00        D      EOS  withdrawl  2.9494
26e+05
2687  59020 2020-02-25 11:28:00        C      BTC    deposit  7.3473
16e+05
2688  74925 2020-02-25 04:27:00        C      EOS    deposit  1.0000
00e+05
2689  24909 2020-02-25 12:37:00        A      ETH    deposit  2.0000
00e+05

                flag
0
1
2
3
4
...              ...
2685  started b4 8am
2686
2687  started b4 8am
2688
2689  started b4 8am

[2690 rows x 7 columns]
```

In [13]:
```
# Flag any transfer that finished after 8am

df['flag'] = np.where(df['time']>'2/25/20 20:00','finished after 8pm',
'')
```

In [14]:
```
print(df)
```

```
      tx_id                time exchange currency    tx_type
Amount  \
0     94719 2020-02-25 07:23:00        B      BTC    deposit  6.2080
00e+05
1     47648 2020-02-25 01:05:00        C      BCH    deposit  1.1492
00e+06
2     64351 2020-02-25 06:09:00        D      EOS    deposit  2.3090
00e+05
3      1720 2020-02-25 01:27:00        B      ETH    deposit  1.2291
87e+06
4     29500 2020-02-25 03:35:00        C      BCH  withdrawl  9.8730
00e+05
...     ...                 ...      ...      ...         ...
...
2685  94252 2020-02-25 13:20:00        B      USD  withdrawl  1.4247
00e+06
2686  14922 2020-02-25 01:05:00        D      EOS  withdrawl  2.9494
26e+05
2687  59020 2020-02-25 11:28:00        C      BTC    deposit  7.3473
16e+05
2688  74925 2020-02-25 04:27:00        C      EOS    deposit  1.0000
00e+05
2689  24909 2020-02-25 12:37:00        A      ETH    deposit  2.0000
00e+05

     flag
0
1
2
3
4
...   ...
2685
2686
2687
2688
2689

[2690 rows x 7 columns]
```

In [15]:
```
# Match pairs of withdrawals and deposits as 'transfers',;

df.groupby('tx_id').agg(lambda x: x.tolist())
```

Out[15]:

tx_id	time	exchange	currency	tx_type	Amount	flag
26	[2020-02-25 11:04:00, 2020-02-25 11:04:00]	[D, B]	[ETH, ETH]	[withdrawl, deposit]	[1926700.0, 1926700.0]	[,]
148	[2020-02-25 11:16:00]	[C]	[BTC]	[withdrawl]	[529400.0]	[]
157	[2020-02-25 12:28:00]	[A]	[USD]	[deposit]	[1125687.655]	[]
229	[2020-02-25 05:44:00]	[D]	[EOS]	[deposit]	[1191300.0]	[]
339	[2020-02-25 07:14:00]	[D]	[USD]	[withdrawl]	[1842200.0]	[]
...	...	...	...	...	...	...
99808	[2020-02-25 06:37:00, 2020-02-25 06:37:00]	[D, B]	[USD, USD]	[deposit, withdrawl]	[810700.0, 810700.0]	[,]
99810	[2020-02-25 12:28:00, 2020-02-25 12:28:00]	[B, D]	[BCH, BCH]	[deposit, withdrawl]	[1298206.332, 1298206.332]	[,]
99854	[2020-02-25 11:23:00]	[D]	[BCH]	[deposit]	[1637200.0]	[]
99926	[2020-02-25 11:11:00]	[D]	[BTC]	[deposit]	[221200.0]	[]
99937	[2020-02-25 06:16:00]	[A]	[ETH]	[deposit]	[286300.0]	[]

1926 rows × 6 columns

In [ ]:
```
def twosum(self, n)
```