Тестовое задание

In [35]:
```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [36]:
```
df = pd.read_csv('Dataset_points.csv', sep = '\t')
```

Для начала посмотрим, как выглядят наши данные

In [21]:
```
df.head()
```

Out[21]:

	period	recordkind	customerid	points
0	2010-03-31 00:00:00	0	0540fe8a-c06e-42f4-8015-dd874f7ec443	8511.757
1	2010-03-31 00:00:00	0	5750e29a-e783-460a-8031-30149f8e302b	329.994
2	2010-03-31 00:00:00	0	5a24042b-fff7-4a36-8025-03e2d1e14ef2	449.960
3	2010-03-31 00:00:00	0	741912b5-583b-4785-8040-82345d8610cb	115.213
4	2010-04-02 11:47:08	0	0540fe8a-c06e-42f4-8015-dd874f7ec443	64.000

Создадим цикл, который сделает все поинты, относящиеся к расходу, отрицательными

In [37]:
```
for index, row in df.iterrows():
    if row['recordkind'] == 1:
        df['points'][index] = row['points']*(-1)
```

Далее возьмем переменную period и приведем ее к формату pandas для дальнейших манипуляций. Затем отсортируем наши данные по дате для наглядности

In [38]:
```
df['period'] =pd.to_datetime(df['period'])
df.sort_values(by='period')
```

Out[38]:

	period	recordkind	customerid	points
0	2010-03-31 00:00:00	0	0540fe8a-c06e-42f4-8015-dd874f7ec443	8511.757
9791	2010-03-31 00:00:00	0	dc291d2b-1ae7-4a54-803a-032a46243584	1150.650
9788	2010-03-31 00:00:00	0	ce57503e-c1c0-476d-803e-385464dfd487	1096.184
9787	2010-03-31 00:00:00	0	347f411c-076c-4b90-8039-25836f10a003	573.854
9790	2010-03-31 00:00:00	0	daee3185-f74a-4b8f-8022-48962097f4fa	3060.251
...	...	...	...	...
19492	2019-11-04 21:01:36	0	27d2f6b4-603a-11e4-89d5-00259038e9f2	7.170
9785	2019-11-05 13:14:58	1	8407e59f-f355-4cba-8035-1f13cc4b1362	-466.000
19493	2019-11-05 15:39:44	1	c0b59565-730f-11e4-89d5-00259038e9f2	-370.500
9786	2019-11-05 21:00:17	0	d4251117-44b2-4d08-8017-3ad4cb40aac1	5.390
19494	2019-11-05 21:16:43	0	d4251117-44b2-4d08-8017-3ad4cb40aac1	0.540

19495 rows × 4 columns

Создадим диапазон дат из наших данных (начало каждого месяца с апреля 2010 по декабрь 2019). Затем создадим цикл, в котором суммарно подсчитываются очки каждого клиента к началу каждого месяца из выбранного нами диапазона.

In [39]:
```
pd.date_range(start='2010-04-01', end='2019-12-01',freq = 'MS')
```

Out[39]:
```
DatetimeIndex(['2010-04-01', '2010-05-01', '2010-06-01', '2010-07-01',
               '2010-08-01', '2010-09-01', '2010-10-01', '2010-11-01',
               '2010-12-01', '2011-01-01',
               ...
               '2019-03-01', '2019-04-01', '2019-05-01', '2019-06-01',
               '2019-07-01', '2019-08-01', '2019-09-01', '2019-10-01',
               '2019-11-01', '2019-12-01'],
              dtype='datetime64[ns]', length=117, freq='MS')
```

In [40]:
```
dff = pd.DataFrame()
for date in pd.date_range(start='2010-04-01', end='2019-12-01',freq = 'MS'):
    df2 = df[df['period'] <= date].groupby('customerid')['points'].agg('sum').
reset_index()
    df2['date'] = date
    dff = dff.append(df2)
```

Посмотрим, что у нас получилось:

In [41]:
```
dff
```

Out[41]:

	customerid	points	date
0	0540fe8a-c06e-42f4-8015-dd874f7ec443	8.511757e+03	2010-04-01
1	347f411c-076c-4b90-8039-25836f10a003	5.738540e+02	2010-04-01
2	5750e29a-e783-460a-8031-30149f8e302b	3.299940e+02	2010-04-01
3	5a24042b-fff7-4a36-8025-03e2d1e14ef2	4.499600e+02	2010-04-01
4	741912b5-583b-4785-8040-82345d8610cb	1.152130e+02	2010-04-01
...	...	...	...
164	f92523bd-970e-4b75-8018-1edad557ff1e	1.330000e+01	2019-12-01
165	f9bde700-68f1-48a4-801f-f3921c963b6a	3.090000e+00	2019-12-01
166	fa4a5d54-eac4-11e5-8530-00259038e9f2	1.418490e+03	2019-12-01
167	fde714d6-e05c-11e5-8530-00259038e9f2	3.810600e+02	2019-12-01
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-12-01

10835 rows × 3 columns

Мы знаем, что у нас в таблице есть пользователи, у которых 0 поинтов. Посмотрим, сколько их

In [42]:
```
print(dff[dff['points'] == 0])
```

```
                                customerid  points       date
51   f07cd557-8cc2-4069-8030-5f3e66686fc0     0.0 2012-09-01
54   f9182d27-3ee1-487e-803c-cf8ea106c82b     0.0 2012-09-01
51   f07cd557-8cc2-4069-8030-5f3e66686fc0     0.0 2012-10-01
54   f9182d27-3ee1-487e-803c-cf8ea106c82b     0.0 2012-10-01
51   f07cd557-8cc2-4069-8030-5f3e66686fc0     0.0 2012-11-01
..                                    ...     ...        ...
146  ddf2ab0b-56ac-4d94-8014-7e2322cf60c9     0.0 2019-12-01
147  de69c6bf-5c05-11e4-89d5-00259038e9f2     0.0 2019-12-01
152  e7b639ac-dc34-46f2-804c-e7b88d65992d     0.0 2019-12-01
156  f14c9ce7-7175-46c8-8013-ce254824d2d9     0.0 2019-12-01
161  f8e1b668-e983-4cb6-8035-4a40897d1183     0.0 2019-12-01

[758 rows x 3 columns]
```

И уберем из нашего датасета этих пользователей (их всего 758)

In [43]:
```
dff = dff[dff.points != 0.0]
```

Посмотрим теперь на наши данные:

In [44]:
```
dff
```

Out[44]:

	customerid	points	date
0	0540fe8a-c06e-42f4-8015-dd874f7ec443	8.511757e+03	2010-04-01
1	347f411c-076c-4b90-8039-25836f10a003	5.738540e+02	2010-04-01
2	5750e29a-e783-460a-8031-30149f8e302b	3.299940e+02	2010-04-01
3	5a24042b-fff7-4a36-8025-03e2d1e14ef2	4.499600e+02	2010-04-01
4	741912b5-583b-4785-8040-82345d8610cb	1.152130e+02	2010-04-01
...	...	...	...
164	f92523bd-970e-4b75-8018-1edad557ff1e	1.330000e+01	2019-12-01
165	f9bde700-68f1-48a4-801f-f3921c963b6a	3.090000e+00	2019-12-01
166	fa4a5d54-eac4-11e5-8530-00259038e9f2	1.418490e+03	2019-12-01
167	fde714d6-e05c-11e5-8530-00259038e9f2	3.810600e+02	2019-12-01
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-12-01

10077 rows × 3 columns

На всякий случай посмотрим, есть ли теперь пользователи без очков

In [45]:
```
print(dff[dff['points'] == 0])
```

```
Empty DataFrame
Columns: [customerid, points, date]
Index: []
```

Видим, что теперь у нас в датасете нет пользователей с 0 поинтами. Далее сделаем сортировку по датам и пользователям для наглядности

In [54]:
```
dff['date'] = pd.to_datetime(dff['date'])
dff = dff.sort_values(by=['customerid', 'date'])
```

Наконец, посмотрим, что у нас получилось в результате:

In [55]:
```
dff
```

Out[55]:

	customerid	points	date
0	000bff1f-8e6b-42b7-8018-f4a1e8e25bb1	5.000000e+01	2018-09-01
0	000bff1f-8e6b-42b7-8018-f4a1e8e25bb1	5.000000e+01	2018-10-01
0	000bff1f-8e6b-42b7-8018-f4a1e8e25bb1	5.000000e+01	2018-11-01
0	000bff1f-8e6b-42b7-8018-f4a1e8e25bb1	5.000000e+01	2018-12-01
0	000bff1f-8e6b-42b7-8018-f4a1e8e25bb1	5.000000e+01	2019-01-01
...	...	...	...
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-08-01
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-09-01
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-10-01
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-11-01
168	ff4735c9-df9a-4914-804e-0852226320fe	-1.350031e-13	2019-12-01

10077 rows × 3 columns