```
import pandas as pd

# CSV 파일에서 데이터 불러오기
data = pd.read_csv("C:/yelin/sp/항생제/항생제.csv")

# 'VTHOS_NO'로 묶은 후 'PRSC_YMD' 오름차순으로 정렬
sorted_data = data.groupby('VTHOS_NO', group_keys=False).apply(lambda x: x.sort_values('PRSC_YMD'))

# 'VTHOS_NO'와 'INGR_NM'로 묶은 후 'PRSC_YMD' 오름차순으로 정렬
grouped_sorted_data = sorted_data.groupby(['VTHOS_NO', 'INGR_NM'], group_keys=False).apply(lambda x: x.sort_values('PRSC_YMD'))

# A열에 1부터 쭉 번호 매기기
grouped_sorted_data['A'] = range(1, len(grouped_sorted_data) + 1)

# 정렬된 데이터를 CSV 파일로 출력
grouped_sorted_data.to_csv("C:/yelin/sp/항생제/정렬된_항생제.csv", index=False)
```

```
import pandas as pd

# CSV 파일에서 데이터 불러오기
data = pd.read_csv("C:/yelin/sp/항생제/정렬된_항생제.csv")

# 'VTHOS_NO'로 데이터 묶기
grouped = data.groupby('VTHOS_NO')

# 'RGNO'가 다른 경우를 찾아 출력
for vthos_no, group_data in grouped:
    unique_rgno_count = group_data['RGNO'].nunique()

    if unique_rgno_count > 1:
        print(f"VTHOS_NO: {vthos_no} 에서 RGNO가 다른 경우가 있습니다.")
```

항생제 파일 날짜 세기

```
import pandas as pd

# CSV 파일에서 데이터 불러오기
data = pd.read_csv("C:/yelin/sp/항생제/정렬된_항생제.csv")

# PRSC_YMD를 "mm-dd" 형식으로 변경
data['PRSC_YMD'] = data['PRSC_YMD'].astype(str).str.zfill(8)  # 8자리로 맞추기
data['PRSC_YMD'] = data['PRSC_YMD'].str[4:6] + '-' + data['PRSC_YMD'].str[6:]  # 변환

# 수정된 데이터를 CSV 파일로 출력
data.to_csv("C:/yelin/sp/항생제/modified_a.csv", index=False)

# 결과 확인
print(data['PRSC_YMD'])
```

```
0         01-03
1         06-22
2         06-23
3         06-24
4         06-25
          ...
423571    04-26
423572    04-06
423573    03-21
423574    08-16
423575    08-17
Name: PRSC_YMD, Length: 423576, dtype: object
```

```
# 특정 열들이 모두 같은 행의 개수를 세어 새로운 열 'count' 추가
data['count'] = data.groupby(['VTHOS_NO', 'RGNO', 'INGR_NM', 'PRSC_CAPA', 'PRSC_NT', 'PRSC_DCNT', 'IMPL_CAPA', 'BS_CTQTY'])['VTHOS_NO'].transform('siz
```

```
# 수정된 데이터를 CSV 파일로 출력
data.to_csv("C:/yelin/sp/항생제/modified_a1.csv", index=False)
```

```
import pandas as pd

# CSV 파일에서 데이터 불러오기
data = pd.read_csv("C:/yelin/sp/항생제/modified_a1.csv")

# count가 2 이상인 경우 중복된 행 중 첫 번째 행만 남기기
filtered_data = data[data['count'] >= 2].drop_duplicates(subset=['VTHOS_NO', 'RGNO', 'INGR_NM', 'PRSC_CAPA', 'PRSC_NT', 'PRSC_DCNT', 'IMPL_CAPA', 'BS_
```

```
# count가 1인 경우는 그대로 유지
filtered_data = pd.concat([filtered_data, data[data['count'] == 1]])

# 'A' 열을 오름차순으로 정렬
sorted_data = filtered_data.sort_values('A')

# count1 열 생성 및 값 계산
sorted_data['count1'] = sorted_data['PRSC_DCNT'] * sorted_data['count']

# 정리된 데이터를 CSV 파일로 출력
sorted_data.to_csv("C:/yelin/sp/항생제/정리된_항생제.csv", index=False)
```

날짜

```
import pandas as pd

# CSV 파일 읽어오기
df_b = pd.read_csv(r'C:/yelin/sp/항생제/정리된_항생제.csv')

# IMPL_CAPA 값을 비교하여 가장 큰 값을 가지는 행 선택
df_b = df_b.sort_values(by='IMPL_CAPA', ascending=False).drop_duplicates(
    subset=['VTHOS_NO', 'RGNO', 'INGR_NM'], keep='first'
)

# 결과를 새로운 파일로 저장
df_b.to_csv(r'C:/yelin/sp/항생제/정리된_항생제c.csv', index=False)
```

```
import pandas as pd

# CSV 파일 읽어오기
df_b = pd.read_csv(r'C:/yelin/sp/항생제/정리된_항생제.csv')
df_c = pd.read_csv(r'C:/yelin/sp/항생제/정리된_항생제c.csv')

# 두 데이터프레임을 VTHOS_NO, RGNO, PRSC_CD, MD_NM, INGR_NM 열을 기준으로 병합
merged_df = pd.merge(df_b, df_c, on=['VTHOS_NO', 'RGNO', 'INGR_NM'], suffixes=('_b', '_c'))

# IMPL_CAPA_b / IMPL_CAPA_c 값을 계산하여 새로운 열 추가 (반올림)
merged_df['IMPL_CAPA_RATIO'] = (merged_df['IMPL_CAPA_b'] / merged_df['IMPL_CAPA_c']).round(2)

# 계산한 값을 정리된_항생제b.csv 파일에 추가하여 저장
df_b['IMPL_CAPA_RATIO'] = merged_df['IMPL_CAPA_RATIO']

# 결과를 정리된_항생제d.csv 파일로 저장
df_b.to_csv(r'C:/yelin/sp/항생제/정리된_항생제d.csv', index=False)
```

```
import pandas as pd

# CSV 파일 읽어오기
df_d = pd.read_csv(r'C:/yelin/sp/항생제/정리된_항생제d.csv')

# IMPL_CAPA_RATIO 값이 1 미만인 경우 count1 열에 IMPL_CAPA_RATIO 값을 곱해줌
df_d.loc[df_d['IMPL_CAPA_RATIO'] < 1, 'count1'] *= df_d['IMPL_CAPA_RATIO']

# IMPL_CAPA_RATIO 열 삭제
df_d1 = df_d.drop(columns=['IMPL_CAPA_RATIO'])

# 결과를 정리된_항생제d1.csv 파일로 저장
df_d1.to_csv(r'C:/yelin/sp/항생제/정리된_항생제d1.csv', index=False)
```

```
import pandas as pd

# CSV 파일 읽어오기
df_c = pd.read_csv(r'C:/yelin/sp/항생제/정리된_항생제c.csv')
df_d1 = pd.read_csv(r'C:/yelin/sp/항생제/정리된_항생제d1.csv')

# 'VTHOS_NO', 'RGNO', 'INGR_NM' 값이 같은 행들을 그룹화하고 count1 값을 합침
grouped = df_d1.groupby(['VTHOS_NO', 'RGNO', 'INGR_NM'])['count1'].sum().reset_index()

# 정리된_항생제c.csv 파일에 새 열 'count1'을 추가하고 해당 값을 초기화
df_c['count1'] = 0

# 정리된_항생제c.csv 파일의 각 행에 대해 해당 조건에 맞는 값을 찾아서 'count1' 값을 업데이트
for index, row in df_c.iterrows():
    match = grouped[
        (grouped['VTHOS_NO'] == row['VTHOS_NO']) &
        (grouped['RGNO'] == row['RGNO']) &
        (grouped['INGR_NM'] == row['INGR_NM'])
```

```
        ]

        if not match.empty:
            df_c.at[index, 'count1'] = match.iloc[0]['count1']

    # 결과를 정리된_항생제g.csv 파일로 저장
    df_c.to_csv(r'C:/yelin/sp/항생제/정리된_항생제d2.csv', index=False)


    import pandas as pd

    # CSV 파일 읽어오기
    input_path_d2 = r'C:/yelin/sp/항생제/정리된_항생제d2.csv'
    df_d2 = pd.read_csv(input_path_d2)

    # 필요 없는 열들을 제거
    columns_to_remove = ['ESB_TX_ID', 'ESB_STATE_CD', 'ESB_TIME', 'count']
    df_d2_cleaned = df_d2.drop(columns=columns_to_remove)

    # 'A' 항목을 오름차순으로 정렬
    df_d2_sorted = df_d2_cleaned.sort_values(by='A', ascending=True)

    # count1 열의 변수명을 DAY로 변경
    df_d2_sorted.rename(columns={'count1': 'DAY'}, inplace=True)

    # IMPL_CAPA 열의 변수명을 TPRSC_CAPA로 변경
    df_d2_sorted.rename(columns={'IMPL_CAPA': 'TPRSC_CAPA'}, inplace=True)

    # 결과를 새로운 파일로 저장
    output_path_d2_sorted = r'C:/yelin/sp/항생제/정리된_항생제d2.csv'
    df_d2_sorted.to_csv(output_path_d2_sorted, index=False)


    import pandas as pd

    # CSV 파일 읽어오기
    input_path_d2 = r'C:/yelin/sp/항생제/정리된_항생제d2.csv'
    df_d2 = pd.read_csv(input_path_d2)

    # 필요 없는 열들을 제거
    columns_to_remove = ['PRSC_CD', 'MD_NM', 'PRSC_CAPA', 'PRSC_NT', 'PRSC_DCNT', 'BS_CTQTY', 'PRSC_YMD',
                         'FRGN_ADMS_CD', 'LTHT_DPRSC_YN']
    df_d2_cleaned = df_d2.drop(columns=columns_to_remove)

    # 'A' 항목을 오름차순으로 정렬
    df_d2_sorted = df_d2_cleaned.sort_values(by='A', ascending=True)

    # count1 열의 변수명을 DAY로 변경
    df_d2_sorted.rename(columns={'count1': 'DAY'}, inplace=True)

    # IMPL_CAPA 열의 변수명을 TPRSC_CAPA로 변경
    df_d2_sorted.rename(columns={'IMPL_CAPA': 'TPRSC_CAPA'}, inplace=True)

    # 결과를 새로운 파일로 저장
    output_path_d2_sorted = r'C:/yelin/sp/항생제/정리된_항생제d3.csv'
    df_d2_sorted.to_csv(output_path_d2_sorted, index=False)
```