Q1 Report

The first object of the question is to determine the type of data input and process them differently, so that's what I did:

If the fifth string of the date input is "-," then we assume the input comes in as "2018-2-3" format, and we use the **strptime** function in Python to create the DateTime object from the string.

Similarly, for the date that has the fifth string of "Q", we assume it is the quarterly date since there is no month that starts with Q. For a date that has only "A" after the year, we assume it is the annual period. For a date that has two alphabetical letters in a row, we assume it is the monthly period.

Another important thing to note about DateTime is the **DLS (daylight saving)**. For markets with daylight saving, they lose an hour on the second Sunday in March and gain an hour on the first Sunday in November. We need to adjust the flat hours accordingly on these days (thus, off-peak and peak hours will be affected, as well). At the same time, since 7x8 contains non HE7 to HE22 hours, DLS would impact 7x8 total hours as well, so we need to make the change there.

The next thing we need to consider is what are considered **peak days**. Based on my search for the Eastern markets, we only celebrate six holidays: "New Year's Day," 'Memorial Day', 'Independence Day,' 'Labor Day,' 'Thanksgiving,' and 'Christmas Day.' These holidays and all weekends are "off-peak days," and all other days are peak days. The Western markets' peak days are the Eastern peak days plus Saturdays. The numply busday_count() function can help us count the peak days perfectly.

Having made all the corner case adjustments, we can calculate hours for each kind of peak type. In general:

Flat hours = total number of days *24 + DLS change (+1 or -1)
7X8 = total number of days*8 + DLS change
On peak = peak days *16, no need to worry about DLS change since DLS happens at 2 am, which is non-peak.
Offpeak = flat hours – on peak
2X16H = (total_days-peak_days)*16H