**Task 1** - **Option Quote Parsing:**

As standard across the industry quotes for products are sent via real-time feeds and files. Especially in fixed income and the OTC market these quotes are not standardized and vary drastically amongst brokers. We would like you to apply this by writing a parsing function in python that will read each message in the enclosed zip file that begin with ‘hycdx\_option\_quotes\_N.txt’ files and extract out the option prices on the High Yield Credit Default Swap Index (HYCDX) given a time stamp and index reference price for each option’s expiry and strike detail. Note the different ways each unique sender will organize the pricing information. This is indicative on how various brokers send quotes that we need to process.

Notice the difference in the terms ‘Expiry’, ‘EXPIRY’ or ‘Exp’ to identify the option expiration differs in each file. As another example the terms ‘REF’ and ‘Ref’ to identify what underlying reference price the options are priced off of. We suggest you use regular expressions to help account for these differences. Most senders will put all prices for options across a single strike on one line. The broker in ‘hycdx\_option\_quotes\_4.txt’ does not follow this format where each line in the message has prices for a put or call option on a single strike.

Prices are not normalized, some brokers represent them in cents and others in dollar terms. Please normalize all prices in cents so that we can adjust all to a single reference underlying price and compare across brokers. Also notice that if you want to use Call / Put then that translates to PAY=Put and RCV=Call.

Ideally we are looking for an elegant generic solution that can handle the multiple file formats. As you are probably aware, columns may shift or additional formats may be introduced from additional brokers. So a parser that can easily adapt and handle these quotes will be the most flexible. Please comment your code as you develop so we can follow your logic.

Store the following below fields from each message. You do not need to perform any calculation or validation of the fields, that would be an extension in the real-world, but for this exercise you may skip. Some of the fields for a given quote are not present so you can simply leave them blank in your structure. As an example some quotes only have a bid/ask, or only have a quote for the call option or put. The desired output is also attached in ‘task1\_output.xlsx’. We understand this is an academic and interview exercise, so exact matches and results may differ slightly. We are most focused on how your explain your approach and design the solution.

Files:

**Quote files to parse**

hycdx\_option\_quotes\_1.txt

hycdx\_option\_quotes\_2.txt

hycdx\_option\_quotes\_3.txt

hycdx\_option\_quotes\_4.txt

**Expected Output**

task1\_output.xlsx

**Fields**

Time – date and datetime

Expiration

Strike px

Strike spd

Option type

Bid price – normalize all prices so they are in cents

Ask price – normalize all prices so they are in cents

Delta

Firm sender

Implied vol – spread

Implied vol – bps

Implied vol – price

**Task 2 – Beta Calculation:**

Implement a function for beta calculation with respect to S&P 500 index (SPY US Equity) for the stock prices provided in the ‘task2\_stock\_data.xlsx’ excel attachment. You can use any standard packages available for this purpose. To reduce outliers please implement winsorization on the returns before using the data in beta calculation. You can use the [following function](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mstats.winsorize.html) to winsorize the data with a 5% limit on both right and left tails.

The beta calculation should use ‘as of date’, ‘beta calculation window’ and ‘observation frequency’ as inputs. For example: we should be able to calculate beta(s) for as of date = ‘2021/10/31’, window = ‘1y’ and observation frequency = ‘weekly’ returns for the data shared in the file. Other inputs for the ‘frequency’ parameter can be: Daily, Weekly, Bi-weekly, Monthly or Quarterly.

Expected output from the function would be a list or dictionary of beta value corresponding to each stock ticker in the data.Please provide us your output for the following scenarios:

asofdate=2021-10-31, window=1y, frequency=daily

asofdate=2021-10-31, window=1y, frequency=weekly

**Input data description:**

We are sharing stock prices data for stock tickers along with the associate SPX (SPY) index's total returns in the ’task2\_stock\_data.xlsx’ attachment.

Similar to Task 1 your results may differ slightly and we are most interested in how you document your approach with comments and the overall design of your code.