CUSIP Generator

Background

A financial security may be defined by different identifiers across multiple data vendors, exchanges and order management systems.

Bloomberg Ticker

A Bloomberg ticker is a string of characters used to identify a financial security in Bloomberg. In the case of futures contracts, they can be represented by:

Table 1: Parts of Bloomberg futures tickers

Ticker Prefix	1 to 3 uppercase alphas
Expiration Month	As per mapping table below
Expiration Year	1 or 2 numerics, e.g. 2017 may be represented by either 7 or 17
Sector	Index or Comdty

Table 2: Futures expiry month mapping table

Month	Code
Jan	F
Feb	G
Mar	H
Apr	J
May	K
Jun	M
Jul	N
Aug	Q
Sep	U
Oct	V
Nov	X
Dec	Z

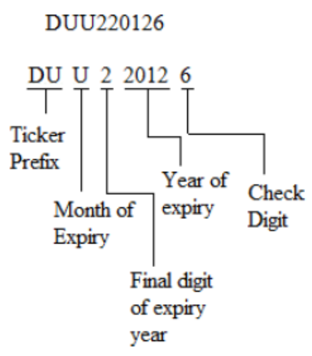
As an example, the WTI Crude Oil Dec 2017 contract is represented by the Bloomberg ticker CLZ7 Comdty.

Modified CUSIP

The standard CUSIP is a nine-character alphanumeric code used to define a North American financial security, consisting of 8 alphanumeric characters (0-9, A-Z, *@#) and a check digit (0-9).

There are 2 approaches to the computation logic for the check digit of a CUSIP. Either with the help of a lookup table, or a fully programmatic approach. A good starting point is:
https://en.wikipedia.org/wiki/CUSIP#Check_digit_lookup_table
https://en.wikipedia.org/wiki/CUSIP#Check_digit_pseudocode

In an order management system (OMS), we have a variant of the standard CUSIP that is used for the specification of futures contracts. The construction logic is:



There are cases where the ticker prefix is only one digit or three digits. In the case of a one digit prefix, repeat the final digit of the year of expiry after the year. In the case of three digit prefixes, the year of expiry will be truncated by the check digit.

The following table provides a list of sample input Bloomberg tickers, how we tokenize them and their corresponding output CUSIPs. This table was generated on 01 Dec 2017:

Input:

Bloomberg Ticker	**AIH8 Index**	**C Z7 Comdty**	**LAZ18 Comdty**	**OATZ7 Comdty**
Ticker Prefix	AI	C	LA	OAT
Expiration Month	H	Z	Z	Z
Expiration Year	2018	2017	2018	2017
Sector	Index	Comdty	Comdty	Comdty

Output:

CUSIP	**AIH820185**	**CZ7201775**	**LAZ820180**	**OATZ72011**

Points to note:

- Bloomberg tickers include either the last or last-two digits of the expiration year in its actual ticker. Assumptions need to be made to resolve a truncated year into the actual expiration year. Taking for example, C Z7 Comdty may be resolved to the 2007, 2017, 2027 expiration contract depending on the run time of this function.
- Bloomberg ticker prefix has a minimum length of 2, so a space is appended for ticker prefixes of length 1 (refer to C Z7 Comdty)

Problem Statement

We have 2 groups of users in our organization that require the generation of modified CUSIPs given Bloomberg Tickers:

i. Users equipped with programming skills like R and Python
ii. Users who prefer to use an interface for interacting with the system to
   a. Convert a single ticker
   b. Convert multiple tickers
   c. Convert a bulk of tickers stored in a flat file

The proposal is to have a RESTful web service along with a web frontend that will satisfy both groups of users.

Requirements

1) Do you agree with this proposal? Why? If not, what is a better solution?
2) Describe the solution in detail, e.g. the choice of programming languages, infrastructure, web server, etc.
3) Create a prototype for the solution that should minimally include the Bloomberg ticker to CUSIP generator. The prototype can also be used to showcase capabilities like client-server interaction and user experience via an interface.
4) State the assumptions used in the prototype that deviate from the actual solution. (e.g. Flask's built-in server was used in the prototype in place of a WSGI server, or that modules were hardcoded or delivered as a concept due to unfamiliarity in that area)

Deliverables

1) Write-ups to questions
2) Source codes
3) Steps to get prototype working if it involves steps beyond running a compiler/interpreter
4) Screenshots of user interface (if available)