**Parsing Functions**

1. **createEntries(mesg, mesgEntries)**

* Function takes **mesg (string)** and **mesgEntries (dictionary {})** as input; **mesg** represents the SWIFT message to parse and **mesgEntries** represents the dictionary to populate with data
* Returns: **mesgEntries** with each block (e.g. 1, 2, 3) and field (e.g. 32A, 50A) representing the key of the dictionary and **mesgEntries[key]** representing the content of each SWIFT message’s field in array form
  + The **mesgEntries** dictionary is meant to resemble a table, with each key representing a column and each index in every array representing the rows

1. **parseMesg(mesg, mesgEntries)**

* Function takes **mesg (string)** and **mesgEntries (dictionary {})** as input; **mesg** represents the rest of the SWIFT message to parse and **mesgEntries** represents the dictionary to populate data with
* Returns: **stringInfo[2]**; this is the result of the method **findNumberAndContent** and it represents the rest of the SWIFT message to be parsed
  + Note: within **createEntries**, **parseMesg** is called within a loop and the result, which is the rest of the SWIFT message to be parsed, is continuously passed back into **parseMesg**

1. **createEntries4(mesg, mesgEntries, fields)**

* Function takes as input **mesg (string)**, **mesgEntries (dictionary {})**, and **fields (array [])** as input; **mesg** represents the content of block 4 to be parsed, **mesgEntries** represents the dictionary to populate with data, and **fields** represents an array of fields within block 4 that correspond to the message type of the particular SWIFT message
* Returns: **mesgEntries** with each block (e.g. 1, 2, 3) and field (e.g. 32A, 50A) representing the key of the dictionary and **mesgEntries[key]** representing the content of each SWIFT message’s field in array form
  + Note: within **parseMesg**, **createEntries4** is called, and this is where **parseMesg** returns

1. **parse4(mesg, mesgEntries, index, fields)**

* Function takes as input **mesg (string)**, **mesgEntries (dictionary {})**, **index (int)**, **fields (array []); mesg** represents the rest of block 4 to parse, **mesgEntries** represents the dictionary to populate data with, **index** represents the field that is currently being looked at, and **fields** represents all of the fields that need to be accounted for with the given message type
* Returns: an array in the form of [**stringInfo[2]** (the rest of block 4 to be parsed), n + 1 (the next index or field to look at)]

1. **findNumberAndContent(text, midChar, endChar, four)**

* Function takes as input **text (string)**, **midChar (char)**, **endChar (char)**, and **four (boolean)**; **text** represents the text (e.g. SWIFT message) to be broken up, **midChar** represents the middle character to break up the string, **endChar** represents the ending character to break up the string, and **four** represents a boolean value indicating if the function is looking at block 4 as **text**
* Returns: an array in the form of [number (e.g. block or field number), content corresponding to the block or field, the rest of the string that has to be parsed]

**createEntries → parseMesg (calls findNumberAndContent) → createEntries4 → parse4 (calls findNumberAndContent)**

**Sample Instructions to Test Parsing Functions**

1. mesg = '{1:F01SWHQBEBBAXXX0000000000}{2:I103PMFAUS66XXXXN}{3:{119:STP}{121:7fae5582-88c9-40b5-927b-39fc18da9d8}}{4#:20:PARSE21053110001#:23B:CRED#:32A:210101EUR100,00#:33B:EUR100,00#:50A:CRLYFRPP#:52A:OTPVRS22#:53A:SWHQBEBB#:59A:GENODEFF120#:70:/INV/40-4120-21-10#:71A:SHA#-}$'
2. mesg = findCov(mesg)[0]
3. mesgEntries = {}
4. mesgEntries = createEntries(mesg, mesgEntries)

**Other Functions**