Assignment the First -- Pseudocode Algorithm

This is **Part 2** of Assignment the First

Goal:

For 2017 BGMP cohort library preps:
- Demultiplex the data
- **Determine the level of index swapping and undetermined index pairs**
- Do for both before and after index read quality filtering

Report # of read-pairs with:
- properly matched indexes per indexes
- index-hopping observed
- unknown indexes

Input

4 FASTQ files -- 2 biological, 2 index
- On Talapas in `/projects/bgmp/shared/2017_sequencing/`
- DON'T MOVE THEM

List of 24 indexes

Output

- 24 FASTQ files with good index pairs
    - i.e. read1 and read2 for 24 index pairs
- 2 FASTQ files with index hopped read-pairs
- 2 FASTQ files for undetermined index pairs
    - i.e. non-matching or low-quality

List of Functions

(**NOTE** include header w/ name & parameters, doc string, tests for each function, return statement -- see example in assignment -- don't need to re-define functions from `bioinfo.py` module)

`convert_phred()` from `bioinfo.py`
- Does this only work for phred33?

```
def compliment(seq: str) -> str:
        '''Returns the compliment of a given sequence.'''
        return compliment
Input: TACCGGAT
Expected output: ATGGCCTA

def get_index_seqs(seq_line: str) -> str:
        '''Get the sequence of the index-pairs from an index FASTQ file. '''
        return index_seqs
Input: TACCGGAT
CTAGCTCA
Expected output: TACCGGAT-CTAGCTCA

def append_index(index: str) -> str:
        '''Append index sequence to end of header for each read-pair in biological read files'''
        return indexed_header
Input: @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
TACCGGAT
CTAGCTCA
Expected output: @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1_TACCGGAT-CTAGCTCA

def demultiplex(indexed_header: str) -> str:
        '''Sorts FASTQ records based on correct index-pairs and sorts remaining reads into groups with
index hopping and unknown indexes.'''
        return index_pair, index_hop, index_unk
Input: @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1_TACCGGAT-CTAGCTCA
Expected output: FASTQ files for each index pair, plus files for collections of index hopping and
unknown indexes
```

Unit Tests

(**NOTE** need test files for each of the 3 categories + test result output FASTQ files)
< add file names & path here >

- Check that `convert_phred()` from `bioinfo.py` works on the phred encoding for these files
  - -Test on short test files with phred33 and phred64
- Check that `compliment()` returns the correct compliment strand and is applied to the appropriate files
- NOTE Illumina machines automatically complement R1 & R3
  - Read1: Insert
  - Read2: Barcode i7 rev. comp.
  - Read3: Barcode i5
  - Read4: Insert rev. comp.
- Verify that `get_index_seqs()` gets the correct index-pairs from the index FASTQ file
  - test with correct and incorrect index-pairs
  - test with index not in list
- Verify that `append_index()` adds the correct index-pairs to the appropriate headers
  - test with correct and incorrect index-pairs
  - test with index not in list
- Check that `demultiplex()` correctly bins each of the 3 index-pair types
  - test with known pairs
  - test with unknown index
  - test with hopped indexes

Pro Tip from Leslie: there's not enough memory to store info from these huge files in lists, etc.
-Read the data, use it, get rid of it

Read the files

- Use `argparse` to get file paths to the **zipped** 4 FASTQ files and list of 24 indexes from the command line as required arguments
- Read in data 4 lines at a time into appropriate variables
  - indicate EOF to prevent infinite loop
- **NOTE** differentiate between the different FASTQ files
  - Need to prevent reading all 4 FASTQs into one looooooong string
  - Need to differentiate between bio reads and index reads
    - Part 1 of assignment defines which file is which
  - Call a series of functions on each file?

Append index seq. to headers##

- Get the index sequences from the index FASTQ files
- Add sequence of index-pair (e.g. AAAAAA-CCCCCC) to the end of each read-pair
  - Check if we've reached end of header line for each bio FASTQ file
  - If so, append index sequence to end
  - Use `get_index_seqs()` and `append_index()`

Convert phred scores

- Need this step b/c we want to see index swapping both before and after index read quality filtering
- Determine if reads are phred -33 or -64
  - See **Part 1**
- Use `convert_phred()` from `bioinfo.py` to convert the scores from ASCII

Demultiplex reads

- If there's an `N` in seq line of either index file
  - put in "unknown" file
  - update "unknown" counter
- Check seq of each index is in list of 24 indexes
  - If not, add to "unknown" file & update "unknown" counter
  - If so, check if it's dual matched i.e. the same at both ends (NOTE from Leslie: there's a twist!)
    - If not, add to "index hopped" file & update "hopped" counter
    - If is, output to corresponding sample file & update the counter for that sample
- Use `demultiplex()`

Report the results

- Use counters that were incremented when sorting the index pairs
- Print to standard out:
  - Number of matched indexes per sample file
  - Number of hopped indexes
  - Number of unknown indexes
- Include brief labels/descriptions to each of the above numbers

- Use a table or tab-separated format?