Demultiplexing, First assignment, Part2

The problem is that we have dual-matched sequencing data where index hopping occurred. So we have some of our sequenced with incorrect indexes, and some with indexes with low quality scores. Our goal is to separate our sequences into 3 categories.

1. Reads that have indexes that are matching (reverse compliments of each other), that are both in our set of barcodes, and are of good quality.
2. Reads that have indexes that are in our sets of barcodes, are of good quality, but are NOT matching.
3. Reads where one or both of the indexes are not in our set of barcodes, or where one or both of them have too low of a Q score.

Four functions that will help with the script

```
def rev_comp(input_string)
        returns reverse compliment of input_string

def add_index_to_header(header, index1, index2)
        returns the header with the indexes added appropriately

def average_phred_score(string)
        returns the average phred score for a given string of PHRED scores

def write_out_record(strings, file)
        all the code to write out our formatted headers to a specific file
```

Some code to add the reverse compliments of our known barcodes to
our set of barcodes

```
forward_barcodes = provided barcodes set from github
reverse_bardcodes = {}

for barcode in forward_barcodes
                reverse_barcodes.add(rev_comp(barcode))
```

Loop through the file, get the record (4 lines) of each file, check if both barcodes exist in our set, if they don't, write the read1 and read2 to their appropriate unknown files. If the indexes do both exist and are reverse compliments of each other, output to the successful reads file. If the barcodes are in the list, and are not reverse compliments, output to index_hopped file. Loop through all records until we hit the EOF, then break.

```
open R1 file, R2 file, R3 file, R4 file:
        While True:
                get read from the 4 files
                index1 = "sequence" from read2 file
                index2 = "sequence" from read3 file
                if index 1 not in forward_barcodes or index 2 not in
reverse_barcodes or if PHRED scores < threshold
                        write read1 and read2 to unknown files
                        count up unknown reads counter
                elif index 1 == rev_comp(index2)
                        write read1 and read2 successful read files
                        count up successful reads counter
                else:
                        write read1 and read2 indexed hop file
                        count up index hopped reads counter

                go to the next read in the file
                if EOF:
                        break
```