```
'''
Lab 1

Trang Van
CIS 41B

Scores class to read in data and print selected data

'''
import collections

class Scores:
    # Function to construct Scores object with a file and dictionary to
process data
    def __init__(self, file):
        self.file = file
        self.scores_dict = dict()


    # Finds and opens the file passed in, reads the files and zips
information
    #  into the scores_dict
    def readFile(self):
        try:
            fin = open(self.file)
        except FileNotFoundError:
            print ("Error: File not found")

        with open(self.file) as fin:
            country = fin.readline().split()
            scores = [line.split() for line in fin]
            self.scores_dict = dict(zip(country, list(zip(*scores))))
                #scores_dict - key: country, value columns of scores
        fin.close()

    #Decorator function to print out the function name of the function
ran, to debug
    def printName(f):
        def wrapper(*args, **kwargs):
            print("Function Name:", str(f).split()[1])
            result = f(*args, **kwargs)
            return result
        return wrapper

    @printName
    # Creates a temp dictionary to sort keys by total score, prints the
countries and info
    #    in ascending order (smallest -> largest)
    def total_scores (self):
        total_dict = {k:sum((int(i) for i in v)) for k,v in
self.scores_dict.items()}
        for key in sorted(total_dict.items(), key=lambda x: x[1]):
            print(key[0], *self.scores_dict[key[0]])
```

```
    # Given limits and abv_bllw by user. Uses any() to print any
    #     country within the score range
    def score_limit (self,limit, abv_bllw = False):
        if abv_bllw is True:
            for k,v in self.scores_dict.items():
                if any(int(i) > int(limit) for i in v):
                    print (k, end=' ')
        else:
            for k,v in self.scores_dict.items():
                if any(int(i) < int(limit) for i in v):
                    print (k, end = ' ')

    @printName
    # Uses a default dictionary to keep count of each score. Prints the
dictionary after.
    def score_frequency(self):
        freq_dict = collections.defaultdict(int)
        for k,v in self.scores_dict.items()
            for i in v:
                freq_dict[i] += 1
        for k,v, in dict(freq_dict).items():
            print('{:>3}: {:>}'.format(k,v))

    # Creates a generator to display a country's info one at a time and
ONLY up until the end
    #   of the data
    def generate_country(self):
        gen_country = (elem for elem in sorted(self.scores_dict.items(),
key = lambda x: x[0]))

        try:
            print(next(gen_country))
        except StopIteration:
            print("End of data")
```