

Homework 3

Math 510

All functions and documentation should be in a single file titled `lastname_firstname_HW03.py` (not an `.ipynb` file) where your last and first name are as they appear on your student ID. The homework is to be completed in Python 3.x. All functions in the homework should be titled and take input as stated in the problem specification.

Your function will only be tested with the input types described in the problem specification. When possible try and reuse functions from earlier questions (as code is intended to be reusable). Pay attention if the specification asks for a **return** or a **print**. Remember to document your code (purpose, input, output), as this will be an important part of your grade. Good luck.

1. The birthday problem explores the probability of at least two people sharing a birthday in a random group of N people. This probability can be calculated exactly with probability theory, but we are interested in simulating the answer. Write a function, **birthday(N)**, that accepts an integer N that represents the size of the room in question. It then simulates 5,000 groups, each of size N , and returns the fraction that have at least two members with the same birthday. You may assume there are 365 birthdays a person could have (we are ignoring leap days) and that they are all equally likely (ignoring seasonal variance in birthrates).
2. Write a recursive function called **binary_search(x, input_list)** that accepts as arguments a number, x , (your function should work for float and integer types) and a list of numbers sorted in ascending order. Use the binary search algorithm (https://en.wikipedia.org/wiki/Binary_search_algorithm), implemented recursively, to return either an index that contains the number (there may be more than one, but you need only return one) or the index where the number would be inserted (i.e. if `binary_search(x, input_list)` returns an index i then `input_list[i] == x` or `input_list[i-1] < x < input_list[i]`, note if x is larger than the largest element in the input list return the length of the list, or if smaller than the smallest return 0).
3. Write a function called `word_lengths()` that accepts a list of words and using a dictionary comprehension returns a dictionary whose keys are integers and values are the set of words from the input list whose length are that of its key. The values should be lower cased. The function should be as follows:

```
def words_of_length(input_list):  
    """    documentation here    """  
    return {dictionary comprehension here}
```

Hint: It might be helpful to use both dictionary and set comprehensions

4. An alternade is a word in which its letters, taken alternatively in a strict sequence, and used in the same order as the original word, make up at least two other words. All letters must be used, but the smaller words are not necessarily of the same length. For example, a word with seven letters where every second letter is used will produce a four-letter word and a three-letter word. Here are two examples:

"board": makes "bad" and "or".

"waists": makes "wit" and "ass".

Using `unixdict.txt` as a sample text file, write a function, **alternade(file_name)**, that accepts as an argument the file name for the dictionary and then goes through each word in the file and tries to make two smaller words using every second letter (to be considered words they also need to be in the `unixdict.txt`). The smaller words **must** also be members of the file so the word `first` would not be included since `frt` and `is` are not **both** words. Return a list of all the alternades (don't worry about what order they are in as your answer will be sorted prior to being checked).

5. Write a function called **checksum(file_name)** that accepts as input a filename. The function takes the text of the file and sums the letters (upper and lowercase letters have the same value) for 'a' = 1, 'b' = 2, 'c' = 3, ... For example 'word' would give 'w' + 'o' + 'r' + 'd' = 23 + 15 + 18 + 4 = 60. The function should return the integer sum of all the letters in the file.