GTx CSE6040x

FA21: Computing for Data Analysis

Help sennethtsz

Course Progress Dates Discussion Wiki

Course / Module 0: Fundamentals (boot... / Supplemental Notebooks: More Python E...

Previous ✓ Next

More Python Exercises - Sample solutions

Bookmark this page

Problem 0

Fast searching in ordered collections. This problem consists of just a single exercise, worth ten (10) points. It is about an elementary principle science algorithm design, which is that one can often do things faster if one exploits structure in the input data.

Suppose you are given a list of **already sorted** numbers.

In [ ]:

```
A = [2, 16, 26, 32, 52, 71, 80, 88]

# These are already sorted:
assert A == sorted(A)
```

Suppose you now want to know whether a certain value exists in this list. A simple way to do that in Python is as follows.

In [ ]:

```
def contains(A, x):
    """Returns True only if the value `x` exists in `A`."""
    return x in A

print("A contains 32: {}".format(contains(A, 32)))
print("A contains 7: {}".format(contains(A, 7)))
print("A contains -10: {}".format(contains(A, -10)))
```

This method works fine and is reasonably fast on small lists. However, if the list is very large, this method can be wasteful, computationally spea

That's because it does **not** take advantage of the fact that `A` is already ordered. In such a case, it should be easier to determine whether the eler (How?)

Exercise 0 (3 + 7 == 10 points). Write a function, `ordered_contains(S, x)`, that takes an **already sorted** list, `S`, as input, and determines whe x. But there is one more condition: your method **must** be **at least ten times faster** than `contains()` for "large" lists!

In particular, there are two test codes for this exercise. The first one checks that your procedure does, indeed, return the correct result by compa to `contains()`, which we will assume is correct. Correctness is worth three (3) points of partial credit out of ten (10). The second test cell check: implementation is faster than `contains()` for a relatively large, but ordered, list. If your implementation is slower for smaller lists, that is okay!

> **Hint.** If you can find a standard Python library routine to help you, by all means, use it!

In [ ]:

```
def ordered_contains(S, x):
    # You may assume that `S` is sorted
### BEGIN SOLUTION
    return ordered_contains__2(S, x)


# Version 0: Use Python's built-in `bisect()` routine
def ordered_contains__0(S, x):
    from bisect import bisect
    i = bisect(S, x)
    return i > 0 and S[i-1] == x

# Version 1: A manual implementation of binary search.
#
# This is the algorithm that this exercise is intended to
# "introduce."
# https://en.wikipedia.org/wiki/Binary_search_algorithm
#
# However, for a variety of implementation reasons, this
# implementation is unlikely to beat Version 0, above.
# Any ideas why not?
#
THRESHOLD__1 = 128 # An "engineering" constant - use to tune speed
def ordered_contains__1(S, x):
    if len(S) <= THRESHOLD__1:
        return contains(S, x)
    midpoint = int(len(S) / 2)
    if x < S[midpoint]:
        return ordered_contains__1(S[:midpoint], x)
    if x > S[midpoint]:
        return ordered_contains__1(S[midpoint+1:], x)
    return True # Found it!

# The following method improves on the above. Why is it faster?
THRESHOLD__2 = 8 # An "engineering" constant - use to tune speed
def ordered_contains__2(S, x, l=0, r=None):
    if r is None: r = len(S)
    if (r-l) <= THRESHOLD__2:
        return contains(S[l:r], x)
    midpoint = int((l+r) / 2)
    if x < S[midpoint]:
        return ordered_contains__2(S, x, l, midpoint)
    if x > S[midpoint]:
        return ordered_contains__2(S, x, midpoint+1, r)
    return True # Found it!
### END SOLUTION

print("A contains 32: {}".format(ordered_contains(A, 32)))
print("A contains 7: {}".format(ordered_contains(A, 7)))
print("A contains -10: {}".format(ordered_contains(A, -10)))
print("\n(Did those results match the earlier example?)")
```

In [ ]:

```
# Test cell: `test_is_correct` (1 point)

from random import randint, sample

def gen_list(n, v_max, v_min=0):
    return sample(range(v_min, v_max), n)

def gen_sorted_list(n, v_max, v_min=0):
    return sorted(gen_list(n, v_max, v_min))

def check_case(S, x):
    msg = "`contains(S, {}) == {}` while `ordered_contains(S, {}) == {}`!"
    true_solution = contains(S, x)
    your_solution = ordered_contains(S, x)
    assert your_solution == true_solution, msg.format(true_solution, your_solution)

S = gen_sorted_list(13, 100)
print("Checking your code on this input: S = {}".format(S))

check_case(S, S[0])
check_case(S, S[0]-1)
check_case(S, S[-1])
check_case(S, S[-1]+1)

for x in gen_list(50, 100, -100):
    check_case(S, x)
print("\n(Passed basic correctness checks.)")

print("\nTiming `contains()`...")
x = randint(-100, 100)
%timeit contains(S, x)

print("\nTiming `ordered_contains()`...")
%timeit ordered_contains(S, x)

print("\n(This problem is small, so it's okay if your method is slower.)")
print("\n(Passed!)")
```

In [ ]:

```
# Test cell: `test_is_faster` (7 points)

N_MIN = 1000000
N_MAX = 2*N_MIN
R_MAX = max(10*N_MAX, 1000000000)

n = randint(N_MIN, N_MAX)
print("Generating a list of size n={}...".format(n))

S_large = gen_sorted_list(n, R_MAX)

print("Quick correctness check...")
x = randint(-R_MAX, R_MAX)
check_case(S_large, x)
print("\n(Passed.)")

print("\nTiming `contains()`...")
t_baseline = %timeit -o contains(S_large, x)
print("\nTiming `ordered_contains()`...")
t_better = %timeit -o ordered_contains(S_large, x)

speedup = t_baseline.average / t_better.average
assert speedup >= 10, "Your method was only {:.2f}x faster (< 1 means it was slower)!".format(speedup)

print("\n(Passed -- you were {:.1f}x faster!)".format(speedup))
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 1

This problem has 2 exercises, worth a total of 10 points.

Exercise 0 (2 points). Write a function, `transform` that takes a string and performs the following operations to it: converts the string to lower cas and removes any spaces in the string.

> In this problem, "space" means only the space character, not other forms of whitespace, such as tabs and newlines.

For example:

```
assert transform('Hello, World!') == 'hello,world'
```

In [ ]:

```
def transform(k):
    ### BEGIN SOLUTION
    return (k.lower().replace(" ", ""))
    ###END SOLUTION

print(transform('Hello, World!'))
```

In [ ]:

```
# Test cell: test_transform

assert transform('HELLO').isupper() is False
assert transform(' ').isspace() is False
assert transform('HeL lO\ttHe\nRe') == 'hello\tthe\nre'

print("\n(Passed!)")
```

Exercise 1 (3 + 5 == 8 points). Write a function, `remove_dups(S)` that takes a list `S` and removes multiple occurrences of any element. The func list with only one occurrence of each element.

For example, `remove_dups(['cat', 'dog', 'sheep', 'dog'])` would return `['cat', 'dog', 'sheep']`.

This exercise has two test cells. The first checks that your implementation returns the right answer. The second tests your solution on a large list credit, your implementation must take less than a certain amount of time!

In [ ]:

```
def remove_dups(S):
###BEGIN SOLUTION
    return remove_dups_1(S)

def remove_dups_0(S): # Correct, but relatively slow on large lists
    newList = []
    for i in S:
        if i not in newList:
            newList.append(i)
    return newList

def remove_dups_1(S): # Should be much faster
    return list(set(S))
###END SOLUTION
```

In [ ]:

```
# Test cell: test_remove_dups_correct
listA = ['cat', 'dog', 'sheep', 'dog']
listB = [1,2,3,4,3,2,1,5]

assert isinstance(remove_dups(listA), list)

your_listA = remove_dups(listA)
assert len(remove_dups(listA)) == 3 and set(your_listA) == {'cat', 'dog', 'sheep'}
your_listB = remove_dups(listB)
assert len(remove_dups(listB)) == 5 and set(your_listB) == set(range(1,6))

print("\n(Passed!)")
```

In [ ]:

```
# Test cell: test_remove_dups_fast

from random import sample
large_list = sample(range(1000000000), 10000)
timing = %timeit -o remove_dups(large_list)
assert timing.average < 1e-2 # 10 ms time limit for 10,000 elements
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 2: Even Fibonacci Terms

> This problem consists of just a single exercise, worth ten (10) points. This problem is the second of many on the site Project Euler (http://www.projecteuler.net), a set of Number Theory-based computational problems. You might find some of the other problems to be f challenges - but be warned, they scale up in difficulty quickly as you complete them!

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with `1` and `1`, the list of terms not exceedin

```
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.
```

That is, the next term, `55+89=144`, exceeds `100`, so we would stop at `89`.

Your computational task is to calculate the sum of the **even-valued** terms that are **less than** a given number. In this example, if the given numbe the sum of even-valued terms would be `2+8+34=44`.

Exercise 0 (10 points). Write a function, `even_fib_sum(n)`, that takes a specific value, `n`, as input, and returns the sum of the even-valued Fibo sequence terms that do not exceed `n`.

For example:

```
even_fib_sum(100) = 44 ( = 2 + 8 + 34)
even_fib_sum(200) = 188 ( = 2 + 8 + 34 + 144)
```

Keep in mind that we are not interested in the **nth** term in the sequence, but simply the Fibonacci numbers that are strictly less than `n`.

In [ ]:

```
def even_fib_sum(n):
    # You may assume that maximum > 0

    ### BEGIN SOLUTION
    a, b, tot = 1, 1, 0
    while b < n:
        if b % 2 == 0:
            tot += b
        a, b = b, a + b
    return tot
    ### END SOLUTION
```

In [ ]:

```
# Test cell: test_even_fib_sum

assert even_fib_sum(2) == 0, "even_fib_sum(2) did not correctly return 0."
assert even_fib_sum(3) == 2, "even_fib_sum(3) did not correctly return 2."
assert even_fib_sum(100) == 44, "even_fib_sum(100) did not correctly return 44."
assert even_fib_sum(1000000) == 1089154, "even_fib_sum(1000000) did not correctly return 1089154."
assert even_fib_sum(4000000) == 4613732, "even_fib_sum(4000000) did not correctly return 4613732."
print("\n(Passed!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 3

This problem has a single exercise worth a total of ten (10) points.

Exercise 0 (10 points). Define a function, `UniqueCharacters(s)`, that given a string `s` returns a tuple of two elements: the first element is the n unique alphabetic characters in the string and the second is the number of unique digits (base-10) in the string.

For example, the string `'ewwwffioj122434'` should output the following tuple: `(6, 4)`. The `6` occurs because there are 6 unique letters (`'e'`, `' 'o'`, and `'j'`) and `4` because there are 4 unique numbers (`'1'`, `'2'`, `'3'`, `'4'`). Special characters may appear in the string but do not count as number.

```
In [ ]: def UniqueCharacters(s):
            ### BEGIN SOLUTION
            return (len(set([i for i in s if i.isalpha()])),
                    len(set([i for i in s if i.isdigit()])))
            ### END SOLUTION
```

```
In [ ]: # Test Cell: 'UniqueCharacters' (10 points)
        assert UniqueCharacters('abc123') == (3,3)
        assert UniqueCharacters('abc') == (3,0)
        assert UniqueCharacters('aaa') == (1,0)
        assert UniqueCharacters('aaa111') == (1,1)
        assert UniqueCharacters('111') == (0,1)
        assert UniqueCharacters('123') == (0,3)
        assert UniqueCharacters('') == (0,0)
        assert UniqueCharacters('///;;;...,????!!!!!###$$$%%%') == (0,0)
        assert UniqueCharacters('//23bd') == (2,2)
        assert UniqueCharacters('b2b3n4s4d9') == (4,4)
        assert UniqueCharacters('9090909p0y90p90y90') == (2,2)
        assert UniqueCharacters('ieowjfiojfioj2342io4ji') == (6,3)
        assert UniqueCharacters('ewwwffioj122434') == (6,4)
        assert UniqueCharacters('dwdj2ru3jf894jgf.,/23,4./3ei8fj2389ej89/,.,./2dd32je98dej89ij') == (9,5)
        print("\n(Passed!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 4

This problem consists of a single exercise worth ten (10) points.

Exercise 0 (10 points). Complete the function `flatten(L)`, which takes a "complex list," `L`, as input and returns a "flat" copy.

By complex, we mean that the input list `L` consists of **arbitrarily nested lists of strings and integers**. For instance, here is a complex list:

```
L = [['a', ['cat'], 2],[[[3]], 'dog'], 4, 5]
```

Observe that there are strings and integers, but that these may be embedded within lists inside lists inside lists...

Given such a list, your computational task is to extract all the strings and integers, and return them in a single flattened list. For example:

```
assert flatten(L) == ['a', 'cat', 2, 3, 'dog', 4, 5]
```

In your flattened version, the strings and integers must appear in the same "left-to-right" order as they do in the original list. For instance, if you s square brackets in the definition of `L` above, then observe that `'cat'` appears to the right of `'a'`, and `2` appears to the right of `'cat'`, and `3` app right of `2`, etc.

> Hint: One reasonable approach to this problem is to use *recursion*, that is, the idea of a function that calls itself to solve subproblems. S "Recursive programs" at the Wikipedia page on Recursion as it is used in computer science (https://en.wikipedia.org/wiki/Recursion_%28computer_science%29#Recursive_programs).

```
In [ ]: def flatten(L):
            assert type(L) is list
```

```
    ### BEGIN SOLUTION
    flatList = []
    for i in L:
        if type(i) is not list:
            flatList += [i]
        else:
            flatList += flatten(i)
    return flatList
    ### END SOLUTION
```

In [ ]:

```
# Test cell: test_flatten (10 points)

L = [['a',['cat'],2],[[[3]],'dog'],4,5]
FL = flatten(L)
True_L = ['a', 'cat', 2, 3, 'dog', 4, 5]
print("Your result: \n{}".format(FL))
print('True result: \n{}'.format(True_L))
assert type(FL) is list and FL == ['a', 'cat', 2, 3, 'dog', 4, 5]
print("\n")

L = [[1,[['b'],2],'t',[[3]],'snow'],'x',['hat',7]]
FL = flatten(L)
True_L = [1, 'b', 2, 't', 3, 'snow', 'x', 'hat', 7]
print("Your result: \n{}".format(FL))
print('True result: \n{}'.format(True_L))
assert type(FL) is list and FL == [1, 'b', 2, 't', 3, 'snow', 'x', 'hat', 7]
print("\n")

L = ['x',1,'z']
FL = flatten(L)
True_L = ['x',1,'z']
print("Your result: \n{}".format(FL))
print('True result: \n{}'.format(True_L))
assert type(FL) is list and FL == ['x',1,'z']
print("\n")

L = []
FL = flatten(L)
True_L = []
print("Your result: \n{}".format(FL))
print('True result: \n{}'.format(True_L))
assert type(FL) is list and FL == []

print("\n(Passed!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c
correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

In [ ]:

Problem 5: Pair counts

This problem consists of a single exercise worth ten (10) points.

Exercise 0 (10 points). Given a list of integers, find the number of pairs that are consecutive.

For example, given the list, `[1, 2, 5, 8]`:

- the pairs that can be formed from the given list are `(1, 2)`, `(1, 5)`, `(1, 8)`, `(2, 5)`, `(2, 8)`, `(5, 8)`;
- the only pair that has consecutive integers is `(1, 2)` and hence the value to be returned is one (1).

If elements in the list are repeated, they should be treated as members of a distinct pair. For instance, if the list were `[1, 1, 1, 2, 2, 5, 8,`
are three ways to choose `1` and two ways to choose `2`, or a total of $3 \times 2 = 6$ ways to choose the pair `(1, 2)`, so that the answer would in this

The first test case below tests for the correctness of the solution whereas the second one tests for the efficiency of the solution. That is, it shoulc
long for the second case to pass! (To get "full credit," try to find a method that takes less than two (2) seconds on the test input of the second tes

> *Application note.* Although this might seem like a toy problem to solve, its application forms the basis of *pattern recognition*. For exampl
> suppose you are trying to discover the buying pattern of products in a supermarket and want to figure out if placing two products next to
> other impact each others' sales. (Does placing milk next to cereal drive both their sales upwards? Or if placing Muesli next to Cereal will

> additional Muesli sales, since people who buy cereal would anyway buy Milk even if it is in the other corner of the store?)
>
> In mapping that concrete placement problem into this abstract analysis problem, you can think of the list of numbers as the shelf numbe products in a receipt, and you are trying to find out the number of cases where the products were in adjacent shelves.

In [ ]:

```
import numpy as np
import CPSol as cp
```

In [ ]:

```
def count_pairs(L):
    assert type(L)==list

    ### BEGIN SOLUTION
    # For one approach, see the "reference" in CPSol.py, included in this repo.

    # Here is a faster method:
    from collections import Counter
    counts = Counter(L)
    unique_items = sorted(counts.keys())
    unique_pairs = zip(unique_items[1:], unique_items[:-1])
    unit_diff_combos = [counts[b]*counts[a] for b, a in unique_pairs if (b-a) == 1]
    return sum(unit_diff_combos)
    ### END SOLUTION
```

In [ ]:

```
# Test cell: Test_Code1

def test_code1():
    L1=[1,2,3,4,5,6,7,8,9]
    L2=[1,1,1,2,2,3,4,10]
    L3=[1,4,7,9]
    L4=[]
    assert count_pairs(L1)==8, "Test Case L1 failed"
    assert count_pairs(L2)==9, "Test Case L2 failed"
    assert count_pairs(L3)==0, "Test Case L3 failed"
    assert count_pairs(L4)==0, "Test Case L4 failed"

    print("\n(Passed!)")

test_code1()
```

In [ ]:

```
# Test cell: Test_Code2

# This test case will test the efficieny of your solution. If it takes too long (>2 min) to run the cod
# please try improving your method.

import numpy as np
biglist = list(np.random.choice(100, 5000, replace=True))

print("Checking correctness on a large, random list...")
result1 = cp.count_pairs_soln(biglist)
result2 = count_pairs(biglist)
assert result1 == result2
print("(Passed correctness check!)")

print("\nChecking speed...")
timing = %timeit -o count_pairs(biglist)
assert timing.average < 2.0
print("(Passed timing check!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 6: Anagram-orama

This problem consists of a single exercise worth ten (10) points.

The code below imports a list of 10,000 common words in the English language and saves it as "commonwordslist".

In [ ]:

```
commonwordslist = ['aa', 'aaa', 'aaron', 'ab', 'abandoned', 'abc', 'aberdeen', 'abilities', 'ability', 
riginal', 'abortion', 'about', 'above', 'abraham', 'abroad', 'abs', 'absence', 'absent', 'absolute', 'a
'absorption', 'abstract', 'abstracts', 'abu', 'abuse', 'ac', 'academic', 'academics', 'academy', 'acc',
'accept', 'acceptable', 'acceptance', 'accepted', 'accepting', 'accepts', 'access', 'accessed', 'access
ccessible', 'accessing', 'accessories', 'accessory', 'accident', 'accidents', 'accommodate', 'accommoda
mmodations', 'accompanied', 'accompanying', 'accomplish', 'accomplished', 'accordance', 'according', 'a
'account', 'accountability', 'accounting', 'accounts', 'accreditation', 'accredited', 'accuracy', 'accu
rately', 'accused', 'acdbentity', 'ace', 'acer', 'achieve', 'achieved', 'achievement', 'achievements', 
'acid', 'acids', 'acknowledge', 'acknowledged', 'acm', 'acne', 'acoustic', 'acquire', 'acquired', 'acqu
cquisitions', 'acre', 'acres', 'acrobat', 'across', 'acrylic', 'act', 'acting', 'action', 'actions', 'a
'activation', 'active', 'actively', 'activists', 'activities', 'activity', 'actor', 'actors', 'actress'
ctual', 'actually', 'acute', 'ad', 'ada', 'adam', 'adams', 'adaptation', 'adapted', 'adapter', 'adapter
e', 'adaptor', 'add', 'added', 'addiction', 'adding', 'addition', 'additional', 'additionally', 'additi
ss', 'addressed', 'addresses', 'addressing', 'adds', 'adelaide', 'adequate', 'adidas', 'adipex', 'adjac
t', 'adjustable', 'adjusted', 'adjustment', 'adjustments', 'admin', 'administered', 'administration', '
ve', 'administrator', 'administrators', 'admission', 'admissions', 'admit', 'admitted', 'adobe', 'adole
pt', 'adopted', 'adoption', 'adrian', 'ads', 'adsl', 'adult', 'adults', 'advance', 'advanced', 'advance
nces', 'advantage', 'advantages', 'adventure', 'adventures', 'adverse', 'advert', 'advertise', 'adverti
vertisements', 'advertiser', 'advertisers', 'advertising', 'advice', 'advise', 'advised', 'advisor', 'a
dvisory', 'advocacy', 'advocate', 'adware', 'ae', 'aerial', 'aerospace', 'af', 'affair', 'affairs', 'af
cted', 'affecting', 'affects', 'affiliate', 'affiliated', 'affiliates', 'affiliation', 'afford', 'affor
hanistan', 'afraid', 'africa', 'african', 'after', 'afternoon', 'afterwards', 'ag', 'again', 'against',
d', 'agencies', 'agency', 'agenda', 'agent', 'agents', 'ages', 'aggregate', 'aggressive', 'aging', 'ago
'agreed', 'agreement', 'agreements', 'agrees', 'agricultural', 'agriculture', 'ah', 'ahead', 'ai', 'aid
'aim', 'aimed', 'aims', 'air', 'aircraft', 'airfare', 'airline', 'airlines', 'airplane', 'airport', 'ai
j', 'ak', 'aka', 'al', 'ala', 'alabama', 'alan', 'alarm', 'alaska', 'albania', 'albany', 'albert', 'alb
m', 'albums', 'albuquerque', 'alcohol', 'alert', 'alerts', 'alex', 'alexander', 'alexandria', 'alfred',
'algeria', 'algorithm', 'algorithms', 'ali', 'alias', 'alice', 'alien', 'align', 'alignment', 'alike', 
l', 'allah', 'allan', 'alleged', 'allen', 'allergy', 'alliance', 'allied', 'allocated', 'allocation', '
owance', 'allowed', 'allowing', 'allows', 'alloy', 'almost', 'alone', 'along', 'alot', 'alpha', 'alphab
pine', 'already', 'also', 'alt', 'alter', 'altered', 'alternate', 'alternative', 'alternatively', 'alte
'although', 'alto', 'aluminium', 'aluminum', 'alumni', 'always', 'am', 'amanda', 'amateur', 'amazing', 
mazoncom', 'amazoncouk', 'ambassador', 'amber', 'ambien', 'ambient', 'amd', 'amend', 'amended', 'amendm
ments', 'amenities', 'america', 'american', 'americans', 'americas', 'amino', 'among', 'amongst', 'amou
s', 'amp', 'ampland', 'amplifier', 'amsterdam', 'amy', 'an', 'ana', 'anaheim', 'anal', 'analog', 'analy
sis', 'analyst', 'analysts', 'analytical', 'analyze', 'analyzed', 'anatomy', 'anchor', 'ancient', 'and'
'anderson', 'andorra', 'andrea', 'andreas', 'andrew', 'andrews', 'andy', 'angel', 'angela', 'angeles', 
nger', 'angle', 'angola', 'angry', 'animal', 'animals', 'animated', 'animation', 'anime', 'ann', 'anna'
nnex', 'annie', 'anniversary', 'annotated', 'annotation', 'announce', 'announced', 'announcement', 'ann
'announces', 'annoying', 'annual', 'annually', 'anonymous', 'another', 'answer', 'answered', 'answering
, 'ant', 'antarctica', 'antenna', 'anthony', 'anthropology', 'anti', 'antibodies', 'antibody', 'anticip
gua', 'antique', 'antiques', 'antivirus', 'antonio', 'anxiety', 'any', 'anybody', 'anymore', 'anyone', 
'anytime', 'anyway', 'anywhere', 'aol', 'ap', 'apache', 'apart', 'apartment', 'apartments', 'api', 'apn
o', 'app', 'apparatus', 'apparel', 'apparent', 'apparently', 'appeal', 'appeals', 'appear', 'appearance
d', 'appearing', 'appears', 'appendix', 'apple', 'appliance', 'appliances', 'applicable', 'applicant', 
, 'application', 'applications', 'applied', 'applies', 'apply', 'applying', 'appointed', 'appointment',
ts', 'appraisal', 'appreciate', 'appreciated', 'appreciation', 'approach', 'approaches', 'appropriate',
ions', 'approval', 'approve', 'approved', 'approx', 'approximate', 'approximately', 'apps', 'apr', 'apr
'aqua', 'aquarium', 'aquatic', 'ar', 'arab', 'arabia', 'arabic', 'arbitrary', 'arbitration', 'arc', 'ar
h', 'architect', 'architects', 'architectural', 'architecture', 'archive', 'archived', 'archives', 'arc
'area', 'areas', 'arena', 'arg', 'argentina', 'argue', 'argued', 'argument', 'arguments', 'arise', 'ari
ona', 'arkansas', 'arlington', 'arm', 'armed', 'armenia', 'armor', 'arms', 'armstrong', 'army', 'arnold
'arrange', 'arranged', 'arrangement', 'arrangements', 'array', 'arrest', 'arrested', 'arrival', 'arriva
e', 'arrived', 'arrives', 'arrow', 'art', 'arthritis', 'arthur', 'article', 'articles', 'artificial', '
tistic', 'artists', 'arts', 'artwork', 'aruba', 'as', 'asbestos', 'ascii', 'ash', 'ashley', 'asia', 'as
e', 'asin', 'ask', 'asked', 'asking', 'asks', 'asn', 'asp', 'aspect', 'aspects', 'aspnet', 'ass', 'assa
bled', 'assembly', 'assess', 'assessed', 'assessing', 'assessment', 'assessments', 'asset', 'assets', '
signed', 'assignment', 'assignments', 'assist', 'assistance', 'assistant', 'assisted', 'assists', 'asso
ociated', 'associates', 'association', 'associations', 'assume', 'assumed', 'assumes', 'assuming', 'ass
ssumptions', 'assurance', 'assure', 'assured', 'asthma', 'astrology', 'astronomy', 'asus', 'at', 'ata',
ens', 'athletes', 'athletic', 'athletics', 'ati', 'atlanta', 'atlantic', 'atlas', 'atm', 'atmosphere', 
c', 'atom', 'atomic', 'attach', 'attached', 'attachment', 'attachments', 'attack', 'attacked', 'attacks
, 'attempted', 'attempting', 'attempts', 'attend', 'attendance', 'attended', 'attending', 'attention', 
'attitudes', 'attorney', 'attorneys', 'attract', 'attraction', 'attractions', 'attractive', 'attribute'
s', 'au', 'auburn', 'auckland', 'auction', 'auctions', 'aud', 'audi', 'audience', 'audio', 'audit', 'au
g', 'august', 'aurora', 'aus', 'austin', 'australia', 'australian', 'austria', 'authentic', 'authentica
or', 'authorities', 'authority', 'authorization', 'authorized', 'authors', 'auto', 'automated', 'automa
atically', 'automation', 'automobile', 'automobiles', 'automotive', 'autos', 'autumn', 'av', 'availabil
able', 'avatar', 'ave', 'avenue', 'average', 'avg', 'avi', 'aviation', 'avoid', 'avoiding', 'avon', 'aw
'awarded', 'awards', 'aware', 'awareness', 'away', 'awesome', 'awful', 'axis', 'aye', 'az', 'azerbaijan
, 'babe', 'babes', 'babies', 'baby', 'bachelor', 'back', 'backed', 'background', 'backgrounds', 'backin
, 'bacon', 'bacteria', 'bacterial', 'bad', 'badge', 'badly', 'bag', 'baghdad', 'bags', 'bahamas', 'bahr
y', 'baker', 'baking', 'balance', 'balanced', 'bald', 'bali', 'ball', 'ballet', 'balloon', 'ballot', 'b
imore', 'ban', 'banana', 'band', 'bands', 'bandwidth', 'bang', 'bangbus', 'bangkok', 'bangladesh', 'ban
g', 'bankruptcy', 'banks', 'banned', 'banner', 'banners', 'baptist', 'bar', 'barbados', 'barbara', 'bar
lona', 'bare', 'barely', 'bargain', 'bargains', 'barn', 'barnes', 'barrel', 'barrier', 'barriers', 'bar
'base', 'baseball', 'based', 'baseline', 'basement', 'basename', 'bases', 'basic', 'basically', 'basics
'basis', 'basket', 'basketball', 'baskets', 'bass', 'bat', 'batch', 'bath', 'bathroom', 'bathrooms', 'b
an', 'batteries', 'battery', 'battle', 'battlefield', 'bay', 'bb', 'bbc', 'bbs', 'bbw', 'bc', 'bd', 'bd
'beach', 'beaches', 'beads', 'beam', 'bean', 'beans', 'bear', 'bearing', 'bears', 'beast', 'beastality'
ty', 'beat', 'beatles', 'beats', 'beautiful', 'beautifully', 'beauty', 'beaver', 'became', 'because', '
comes', 'becoming', 'bed', 'bedding', 'bedford', 'bedroom', 'bedrooms', 'beds', 'bee', 'beef', 'been', 
ore', 'began', 'begin', 'beginner', 'beginners', 'beginning', 'begins', 'begun', 'behalf', 'behavior', 
, 'behaviour', 'behind', 'beijing', 'being', 'beings', 'belarus', 'belfast', 'belgium', 'belief', 'beli
ve', 'believed', 'believes', 'belize', 'belkin', 'bell', 'belle', 'belly', 'belong', 'belongs', 'below'
elts', 'ben', 'bench', 'benchmark', 'bend', 'beneath', 'beneficial', 'benefit', 'benefits', 'benjamin',
'benz', 'berkeley', 'berlin', 'bermuda', 'bernard', 'berry', 'beside', 'besides', 'best', 'bestiality',
'benz', 'berkeley', 'berlin', 'bermuda', 'bernard', 'berry', 'beside', 'besides', 'best', 'bestiality',
s', 'bet', 'beta', 'beth', 'better', 'betting', 'betty', 'between', 'beverage', 'beverages', 'beverly',
'bg', 'bhutan', 'bi', 'bias', 'bible', 'biblical', 'bibliographic', 'bibliography', 'bicycle', 'bid', '
dding', 'bids', 'big', 'bigger', 'biggest', 'bike', 'bikes', 'bikini', 'bill', 'billing', 'billion', 'b
y', 'bin', 'binary', 'bind', 'binding', 'bingo', 'bio', 'biodiversity', 'biographies', 'biography', 'bi
ical', 'biology', 'bios', 'biotechnology', 'bird', 'birds', 'birmingham', 'birth', 'birthday', 'bishop'
tch', 'bite', 'bits', 'biz', 'bizarre', 'bizrate', 'bk', 'bl', 'black', 'blackberry', 'blackjack', 'bla
e', 'blades', 'blah', 'blair', 'blake', 'blame', 'blank', 'blanket', 'blast', 'bleeding', 'blend', 'ble
d', 'blind', 'blink', 'block', 'blocked', 'blocking', 'blocks', 'blog', 'blogger', 'bloggers', 'bloggin
'blond', 'blonde', 'blood', 'bloody', 'bloom', 'bloomberg', 'blow', 'blowing', 'blowjob', 'blowjobs', '
s', 'bluetooth', 'blvd', 'bm', 'bmw', 'bo', 'board', 'boards', 'boat', 'boating', 'boats', 'bob', 'bobb
'bodies', 'body', 'bold', 'bolivia', 'bolt', 'bomb', 'bon', 'bond', 'bondage', 'bonds', 'bone', 'bones'
'boob', 'boobs', 'book', 'booking', 'bookings', 'bookmark', 'bookmarks', 'books', 'bookstore', 'bool',
'boom', 'boost', 'boot', 'booth', 'boots', 'booty', 'border', 'borders', 'bored', 'boring', 'born', 'bo
nia', 'boss', 'boston', 'both', 'bother', 'botswana', 'bottle', 'bottles', 'bottom', 'bought', 'boulder
d', 'bound', 'boundaries', 'boundary', 'bouquet', 'boutique', 'bow', 'bowl', 'bowling', 'box', 'boxed',
oxing', 'boy', 'boys', 'bp', 'br', 'bra', 'bracelet', 'bracelets', 'bracket', 'brad', 'bradford', 'brad
n', 'brake', 'brakes', 'branch', 'branches', 'brand', 'brandon', 'brands', 'bras', 'brass', 'brave', 'b
zilian', 'breach', 'bread', 'break', 'breakdown', 'breakfast', 'breaking', 'breaks', 'breast', 'breasts
'breathing', 'breed', 'breeding', 'breeds', 'brian', 'brick', 'bridal', 'bride', 'bridge', 'bridges', '
efing', 'briefly', 'briefs', 'bright', 'brighton', 'brilliant', 'bring', 'bringing', 'brings', 'brisban
l', 'britain', 'britannica', 'british', 'britney', 'broad', 'broadband', 'broadcast', 'broadcasting', '
roadway', 'brochure', 'brochures', 'broke', 'broken', 'broker', 'brokers', 'bronze', 'brook', 'brooklyn
'bros', 'brother', 'brothers', 'brought', 'brown', 'browse', 'browser', 'browsers', 'browsing', 'bruce'
'brunette', 'brunswick', 'brush', 'brussels', 'brutal', 'bryan', 'bryant', 'bs', 'bt', 'bubble', 'buck'
'budapest', 'buddy', 'budget', 'budgets', 'buf', 'buffalo', 'buffer', 'bufing', 'bug', 'bugs', 'build',
'builders', 'building', 'buildings', 'builds', 'built', 'bukkake', 'bulgaria', 'bulgarian', 'bulk', 'bu
t', 'bulletin', 'bumper', 'bunch', 'bundle', 'bunny', 'burden', 'bureau', 'buried', 'burke', 'burlingto
'burner', 'burning', 'burns', 'burst', 'burton', 'bus', 'buses', 'bush', 'business', 'businesses', 'bus
'but', 'butler', 'butt', 'butter', 'butterfly', 'button', 'buttons', 'butts', 'buy', 'buyer', 'buyers',
'buys', 'buzz', 'bw', 'by', 'bye', 'byte', 'bytes', 'c', 'ca', 'cab', 'cabin', 'cabinet', 'cabinets', '
les', 'cache', 'cached', 'cad', 'cadillac', 'cafe', 'cage', 'cake', 'cakes', 'cal', 'calcium', 'calcula
ated', 'calculation', 'calculations', 'calculator', 'calculators', 'calendar', 'calendars', 'calgary',
n', 'calif', 'california', 'call', 'called', 'calling', 'calls', 'calm', 'calvin', 'cam', 'cambodia', '
'camcorder', 'camcorders', 'came', 'camel', 'camera', 'cameras', 'cameron', 'cameroon', 'camp', 'campai
gns', 'campbell', 'camping', 'camps', 'campus', 'cams', 'can', 'canada', 'canadian', 'canal', 'canberra
'cancellation', 'cancelled', 'cancer', 'candidate', 'candidates', 'candle', 'candles', 'candy', 'cannon
'cant', 'canvas', 'canyon', 'cap', 'capabilities', 'capability', 'capable', 'capacity', 'cape', 'capita
l', 'caps', 'captain', 'capture', 'captured', 'car', 'carb', 'carbon', 'card', 'cardiac', 'cardiff', 'c
r', 'cards', 'care', 'career', 'careers', 'careful', 'carefully', 'carey', 'cargo', 'caribbean', 'carin
'carlo', 'carlos', 'carmen', 'carnival', 'carol', 'carolina', 'caroline', 'carpet', 'carried', 'carrier
s', 'carries', 'carroll', 'carry', 'carrying', 'cars', 'cart', 'carter', 'cartoon', 'cartoons', 'cartri
idges', 'cas', 'casa', 'case', 'cases', 'casey', 'cash', 'cashiers', 'casino', 'casinos', 'casio', 'cas
t', 'casting', 'castle', 'casual', 'cat', 'catalog', 'catalogs', 'catalogue', 'catalyst', 'catch', 'cat
ategory', 'catering', 'cathedral', 'catherine', 'catholic', 'cats', 'cattle', 'caught', 'cause', 'cause
, 'causing', 'caution', 'cave', 'cayman', 'cb', 'cbs', 'cc', 'ccd', 'cd', 'cdna', 'cds', 'cdt', 'ce', '
ling', 'celebrate', 'celebration', 'celebrities', 'celebrity', 'celebs', 'cell', 'cells', 'cellular', '
ment', 'cemetery', 'census', 'cent', 'center', 'centered', 'centers', 'central', 'centre', 'centres', '
turies', 'century', 'ceo', 'ceramic', 'ceremony', 'certain', 'certainly', 'certificate', 'certificates'
tion', 'certified', 'cest', 'cet', 'cf', 'cfr', 'cg', 'cgi', 'ch', 'chad', 'chain', 'chains', 'chair',
'chairs', 'challenge', 'challenged', 'challenges', 'challenging', 'chamber', 'chambers', 'champagne', '
'champions', 'championship', 'championships', 'chan', 'chance', 'chancellor', 'chances', 'change', 'cha
gelog', 'changes', 'changing', 'channel', 'channels', 'chaos', 'chapel', 'chapter', 'chapters', 'char',
, 'characteristic', 'characteristics', 'characterization', 'characterized', 'characters', 'charge', 'ch
rger', 'chargers', 'charges', 'charging', 'charitable', 'charity', 'charles', 'charleston', 'charlie',
'charm', 'charming', 'charms', 'chart', 'charter', 'charts', 'chase', 'chassis', 'chat', 'cheap', 'chea
est', 'cheat', 'cheats', 'check', 'checked', 'checking', 'checklist', 'checkout', 'checks', 'cheers', '
ef', 'chelsea', 'chem', 'chemical', 'chemicals', 'chemistry', 'chen', 'cheque', 'cherry', 'chess', 'che
r', 'chevrolet', 'chevy', 'chi', 'chicago', 'chick', 'chicken', 'chicks', 'chief', 'child', 'childhood'
, 'childrens', 'chile', 'china', 'chinese', 'chip', 'chips', 'cho', 'chocolate', 'choice', 'choices', '
lesterol', 'choose', 'choosing', 'chorus', 'chose', 'chosen', 'chris', 'christ', 'christian', 'christia
stians', 'christina', 'christine', 'christmas', 'christopher', 'chrome', 'chronic', 'chronicle', 'chron
ysler', 'chubby', 'chuck', 'church', 'churches', 'ci', 'cia', 'cialis', 'ciao', 'cigarette', 'cigarette
ati', 'cindy', 'cinema', 'cingular', 'cio', 'cir', 'circle', 'circles', 'circuit', 'circuits', 'circula
tion', 'circumstances', 'circus', 'cisco', 'citation', 'citations', 'cite', 'cited', 'cities', 'citizen
s', 'citizenship', 'city', 'citysearch', 'civic', 'civil', 'civilian', 'civilization', 'cj', 'cl', 'cla
d', 'claims', 'claire', 'clan', 'clara', 'clarity', 'clark', 'clarke', 'class', 'classes', 'classic', '
'classics', 'classification', 'classified', 'classifieds', 'classroom', 'clause', 'clay', 'clean', 'cle
ners', 'cleaning', 'cleanup', 'clear', 'clearance', 'cleared', 'clearing', 'clearly', 'clerk', 'clevela
, 'clicking', 'clicks', 'client', 'clients', 'cliff', 'climate', 'climb', 'climbing', 'clinic', 'clinic
s', 'clinton', 'clip', 'clips', 'clock', 'clocks', 'clone', 'close', 'closed', 'closely', 'closer', 'cl
est', 'closing', 'closure', 'cloth', 'clothes', 'clothing', 'cloud', 'clouds', 'cloudy', 'club', 'clubs
, 'clusters', 'cm', 'cms', 'cn', 'cnet', 'cnetcom', 'cnn', 'co', 'coach', 'coaches', 'coaching', 'coal'
n', 'coast', 'coastal', 'coat', 'coated', 'coating', 'cock', 'cocks', 'cod', 'code', 'codes', 'coding',
'cognitive', 'cohen', 'coin', 'coins', 'col', 'cold', 'cole', 'coleman', 'colin', 'collaboration', 'col
'collapse', 'collar', 'colleague', 'colleagues', 'collect', 'collectables', 'collected', 'collectible',
es', 'collecting', 'collection', 'collections', 'collective', 'collector', 'collectors', 'college', 'co
llins', 'cologne', 'colombia', 'colon', 'colonial', 'colony', 'color', 'colorado', 'colored', 'colors',
'colours', 'columbia', 'columbus', 'column', 'columnists', 'columns', 'com', 'combat', 'combination', '
s', 'combine', 'combined', 'combines', 'combining', 'combo', 'come', 'comedy', 'comes', 'comfort', 'com
'comic', 'comics', 'coming', 'comm', 'command', 'commander', 'commands', 'comment', 'commentary', 'comm
ments', 'commerce', 'commercial', 'commission', 'commissioner', 'commissioners', 'commissions', 'commit
nt', 'commitments', 'committed', 'committee', 'committees', 'commodities', 'commodity', 'common', 'comm
ons', 'commonwealth', 'communicate', 'communication', 'communications', 'communist', 'communities', 'co
omp', 'compact', 'companies', 'companion', 'company', 'compaq', 'comparable', 'comparative', 'compare',
'comparing', 'comparison', 'comparisons', 'compatibility', 'compatible', 'compensation', 'compete', 'co
ompeting', 'competition', 'competitions', 'competitive', 'competitors', 'compilation', 'compile', 'comp
iler', 'complaint', 'complaints', 'complement', 'complete', 'completed', 'completely', 'completing', 'c
'complex', 'complexity', 'compliance', 'compliant', 'complicated', 'complications', 'complimentary', 'c
ponent', 'components', 'composed', 'composer', 'composite', 'composition', 'compound', 'compounds', 'co
```

```
ponent', 'components', 'composed', 'composer', 'composite', 'composition', 'compound', 'compounds', 'co
, 'compressed', 'compression', 'compromise', 'computation', 'computational', 'compute', 'computed', 'co
mputers', 'computing', 'con', 'concentrate', 'concentration', 'concentrations', 'concept', 'concepts',
, 'concern', 'concerned', 'concerning', 'concerns', 'concert', 'concerts', 'conclude', 'concluded', 'co
'conclusions', 'concord', 'concrete', 'condition', 'conditional', 'conditioning', 'conditions', 'condo'
'conduct', 'conducted', 'conducting', 'conf', 'conference', 'conferences', 'conferencing', 'confidence'
t', 'confidential', 'confidentiality', 'config', 'configuration', 'configure', 'configured', 'configuri
m', 'confirmation', 'confirmed', 'conflict', 'conflicts', 'confused', 'confusion', 'congo', 'congratula
gress', 'congressional', 'conjunction', 'connect', 'connected', 'connecticut', 'connecting', 'connectio
ions', 'connectivity', 'connector', 'connectors', 'cons', 'conscious', 'consciousness', 'consecutive',
'consent', 'consequence', 'consequences', 'consequently', 'conservation', 'conservative', 'consider', '
e', 'consideration', 'considerations', 'considered', 'considering', 'considers', 'consist', 'consistenc
ent', 'consistently', 'consisting', 'consists', 'console', 'consoles', 'consolidated', 'consolidation',
m', 'conspiracy', 'const', 'constant', 'constantly', 'constitute', 'constitutes', 'constitution', 'cons
'constraint', 'constraints', 'construct', 'constructed', 'construction', 'consult', 'consultancy', 'con
onsultants', 'consultation', 'consulting', 'consumer', 'consumers', 'consumption', 'contact', 'contacte
ing', 'contacts', 'contain', 'contained', 'container', 'containers', 'containing', 'contains', 'contami
ntemporary', 'content', 'contents', 'contest', 'contests', 'context', 'continent', 'continental', 'cont
ontinue', 'continued', 'continues', 'continuing', 'continuity', 'continuous', 'continuously', 'contract
ing', 'contractor', 'contractors', 'contracts', 'contrary', 'contrast', 'contribute', 'contributed', 'c
, 'contribution', 'contributions', 'contributor', 'contributors', 'control', 'controlled', 'controller'
rs', 'controlling', 'controls', 'controversial', 'controversy', 'convenience', 'convenient', 'conventio
ional', 'conventions', 'convergence', 'conversation', 'conversations', 'conversion', 'convert', 'conver
rter', 'convertible', 'convicted', 'conviction', 'convinced', 'cook', 'cookbook', 'cooked', 'cookie', '
ooking', 'cool', 'cooler', 'cooling', 'cooper', 'cooperation', 'cooperative', 'coordinate', 'coordinate
ates', 'coordination', 'coordinator', 'cop', 'cope', 'copied', 'copies', 'copper', 'copy', 'copying', '
'copyrighted', 'copyrights', 'coral', 'cord', 'cordless', 'core', 'cork', 'corn', 'cornell', 'corner',
'cornwall', 'corp', 'corporate', 'corporation', 'corporations', 'corps', 'corpus', 'correct', 'correcte
ion', 'corrections', 'correctly', 'correlation', 'correspondence', 'corresponding', 'corruption', 'cos'
, 'cosmetics', 'cost', 'costa', 'costs', 'costume', 'costumes', 'cottage', 'cottages', 'cotton', 'could
, 'councils', 'counsel', 'counseling', 'count', 'counted', 'counter', 'counters', 'counties', 'counting
s', 'country', 'counts', 'county', 'couple', 'coupled', 'couples', 'coupon', 'coupons', 'courage', 'cou
se', 'courses', 'court', 'courtesy', 'courts', 'cove', 'cover', 'coverage', 'covered', 'covering', 'cov
'cowboy', 'cox', 'cp', 'cpu', 'cr', 'crack', 'cradle', 'craft', 'crafts', 'craig', 'crap', 'craps', 'cr
ord', 'crazy', 'cream', 'create', 'created', 'creates', 'creating', 'creation', 'creations', 'creative'
y', 'creator', 'creature', 'creatures', 'credit', 'credits', 'creek', 'crest', 'crew', 'cricket', 'crim
, 'criminal', 'crisis', 'criteria', 'criterion', 'critical', 'criticism', 'critics', 'crm', 'croatia',
ps', 'cross', 'crossing', 'crossword', 'crowd', 'crown', 'crucial', 'crude', 'cruise', 'cruises', 'cruz
rystal', 'cs', 'css', 'cst', 'ct', 'cu', 'cuba', 'cube', 'cubic', 'cuisine', 'cult', 'cultural', 'cultu
es', 'cum', 'cumshot', 'cumshots', 'cumulative', 'cunt', 'cup', 'cups', 'cure', 'curious', 'currencies'
, 'current', 'currently', 'curriculum', 'cursor', 'curtis', 'curve', 'curves', 'custody', 'custom', 'cu
stomers', 'customise', 'customize', 'customized', 'customs', 'cut', 'cute', 'cuts', 'cutting', 'cv', 'c
'cyber', 'cycle', 'cycles', 'cycling', 'cylinder', 'cyprus', 'cz', 'czech', 'd', 'da', 'dad', 'daddy',
iry', 'daisy', 'dakota', 'dale', 'dallas', 'dam', 'damage', 'damaged', 'damages', 'dame', 'damn', 'dan'
ance', 'dancing', 'danger', 'dangerous', 'daniel', 'danish', 'danny', 'dans', 'dare', 'dark', 'darkness
'das', 'dash', 'dat', 'data', 'database', 'databases', 'date', 'dated', 'dates', 'dating', 'daughter',
'dave', 'david', 'davidson', 'davis', 'dawn', 'day', 'days', 'dayton', 'db', 'dc', 'dd', 'ddr', 'de', '
line', 'deadly', 'deaf', 'deal', 'dealer', 'dealers', 'dealing', 'deals', 'dealt', 'dealtime', 'dean',
th', 'deaths', 'debate', 'debian', 'deborah', 'debt', 'debug', 'debut', 'dec', 'decade', 'decades', 'de
cent', 'decide', 'decided', 'decimal', 'decision', 'decisions', 'deck', 'declaration', 'declare', 'decl
ine', 'declined', 'decor', 'decorating', 'decorative', 'decrease', 'decreased', 'dedicated', 'dee', 'de
p', 'deeper', 'deeply', 'deer', 'def', 'default', 'defeat', 'defects', 'defence', 'defend', 'defendant'
'defensive', 'deferred', 'deficit', 'define', 'defined', 'defines', 'defining', 'definitely', 'definiti
tions', 'degree', 'degrees', 'del', 'delaware', 'delay', 'delayed', 'delays', 'delegation', 'delete', '
elhi', 'delicious', 'delight', 'deliver', 'delivered', 'delivering', 'delivers', 'delivery', 'dell', 'd
xe', 'dem', 'demand', 'demanding', 'demands', 'demo', 'democracy', 'democrat', 'democratic', 'democrats
hic', 'demonstrate', 'demonstrated', 'demonstrates', 'demonstration', 'den', 'denial', 'denied', 'denma
s', 'dense', 'density', 'dental', 'dentists', 'denver', 'deny', 'department', 'departmental', 'departme
ture', 'depend', 'dependence', 'dependent', 'depending', 'depends', 'deployment', 'deposit', 'deposits'
'depression', 'dept', 'depth', 'deputy', 'der', 'derby', 'derek', 'derived', 'des', 'descending', 'desc
ribed', 'describes', 'describing', 'description', 'descriptions', 'desert', 'deserve', 'design', 'desig
ignation', 'designed', 'designer', 'designers', 'designing', 'designs', 'desirable', 'desire', 'desired
'desktop', 'desktops', 'desperate', 'despite', 'destination', 'destinations', 'destiny', 'destroy', 'de
estruction', 'detail', 'detailed', 'details', 'detect', 'detected', 'detection', 'detective', 'detector
ation', 'determine', 'determined', 'determines', 'determining', 'detroit', 'deutsch', 'deutsche', 'deut
ev', 'devel', 'develop', 'developed', 'developer', 'developers', 'developing', 'development', 'developm
elopments', 'develops', 'deviant', 'deviation', 'device', 'devices', 'devil', 'devon', 'devoted', 'df',
'di', 'diabetes', 'diagnosis', 'diagnostic', 'diagram', 'dial', 'dialog', 'dialogue', 'diameter', 'diam
nds', 'diana', 'diane', 'diary', 'dice', 'dick', 'dicke', 'dicks', 'dictionaries', 'dictionary', 'did',
d', 'diego', 'dies', 'diesel', 'diet', 'dietary', 'diff', 'differ', 'difference', 'differences', 'diffe
erential', 'differently', 'difficult', 'difficulties', 'difficulty', 'diffs', 'dig', 'digest', 'digit',
'dildo', 'dildos', 'dim', 'dimension', 'dimensional', 'dimensions', 'dining', 'dinner', 'dip', 'diploma
irect', 'directed', 'direction', 'directions', 'directive', 'directly', 'director', 'directories', 'dir
rectory', 'dirt', 'dirty', 'dis', 'disabilities', 'disability', 'disable', 'disabled', 'disagree', 'dis
'disaster', 'disc', 'discharge', 'disciplinary', 'discipline', 'disciplines', 'disclaimer', 'disclaimer
e', 'disclosure', 'disco', 'discount', 'discounted', 'discounts', 'discover', 'discovered', 'discovery'
, 'discretion', 'discrimination', 'discs', 'discuss', 'discussed', 'discusses', 'discussing', 'discussi
sions', 'disease', 'diseases', 'dish', 'dishes', 'disk', 'disks', 'disney', 'disorder', 'disorders', 'd
ispatched', 'display', 'displayed', 'displaying', 'displays', 'disposal', 'disposition', 'dispute', 'di
st', 'distance', 'distances', 'distant', 'distinct', 'distinction', 'distinguished', 'distribute', 'dis
'distribution', 'distributions', 'distributor', 'distributors', 'district', 'districts', 'disturbed', '
, 'diverse', 'diversity', 'divide', 'divided', 'dividend', 'divine', 'diving', 'division', 'divisions',
'divx', 'diy', 'dj', 'dk', 'dl', 'dm', 'dna', 'dns', 'do', 'doc', 'dock', 'docs', 'doctor', 'doctors',
'document', 'documentary', 'documentation', 'documentcreatetextnode', 'documented', 'documents', 'dod',
oe', 'does', 'dog', 'dogs', 'doing', 'doll', 'dollar', 'dollars', 'dolls', 'dom', 'domain', 'domains',
estic', 'dominant', 'dominican', 'don', 'donald', 'donate', 'donated', 'donation', 'donations', 'done',
onor', 'donors', 'dont', 'doom', 'door', 'doors', 'dos', 'dosage', 'dose', 'dot', 'double', 'doubt', 'd
as', 'dover', 'dow', 'down', 'download', 'downloadable', 'downloadcom', 'downloaded', 'downloading', 'd
'downtown', 'dozen', 'dozens', 'dp', 'dpi', 'dr', 'draft', 'drag', 'dragon', 'drain', 'drainage', 'dram
```

```
c', 'dramatically', 'draw', 'drawing', 'drawings', 'drawn', 'draws', 'dream', 'dreams', 'dress', 'dress
s', 'dressing', 'drew', 'dried', 'drill', 'drilling', 'drink', 'drinking', 'drinks', 'drive', 'driven',
'drivers', 'drives', 'driving', 'drop', 'dropped', 'drops', 'drove', 'drug', 'drugs', 'drum', 'drums',
y', 'dryer', 'ds', 'dsc', 'dsl', 'dt', 'dts', 'du', 'dual', 'dubai', 'dublin', 'duck', 'dude', 'due', '
, 'dumb', 'dump', 'duncan', 'duo', 'duplicate', 'durable', 'duration', 'durham', 'during', 'dust', 'dut
s', 'duty', 'dv', 'dvd', 'dvds', 'dx', 'dying', 'dylan', 'dynamic', 'dynamics', 'e', 'ea', 'each', 'eag
s', 'ear', 'earl', 'earlier', 'earliest', 'early', 'earn', 'earned', 'earning', 'earnings', 'earrings',
rth', 'earthquake', 'ease', 'easier', 'easily', 'east', 'easter', 'eastern', 'easy', 'eat', 'eating', '
, 'ebony', 'ebook', 'ebooks', 'ec', 'echo', 'eclipse', 'eco', 'ecological', 'ecology', 'ecommerce', 'ec
onomics', 'economies', 'economy', 'ecuador', 'ed', 'eddie', 'eden', 'edgar', 'edge', 'edges', 'edinburg
'edited', 'editing', 'edition', 'editions', 'editor', 'editorial', 'editorials', 'editors', 'edmonton',
t', 'educated', 'education', 'educational', 'educators', 'edward', 'edwards', 'ee', 'ef', 'effect', 'ef
ffectively', 'effectiveness', 'effects', 'efficiency', 'efficient', 'efficiently', 'effort', 'efforts',
, 'eggs', 'egypt', 'egyptian', 'eh', 'eight', 'either', 'ejaculation', 'el', 'elder', 'elderly', 'elect
, 'election', 'elections', 'electoral', 'electric', 'electrical', 'electricity', 'electro', 'electron',
c', 'electronics', 'elegant', 'element', 'elementary', 'elements', 'elephant', 'elevation', 'eleven', '
, 'eligible', 'eliminate', 'elimination', 'elite', 'elizabeth', 'ellen', 'elliott', 'ellis', 'else', 'e
'elvis', 'em', 'emacs', 'email', 'emails', 'embassy', 'embedded', 'emerald', 'emergency', 'emerging', '
nem', 'emirates', 'emission', 'emissions', 'emma', 'emotional', 'emotions', 'emperor', 'emphasis', 'emp
ical', 'employ', 'employed', 'employee', 'employees', 'employer', 'employers', 'employment', 'empty', '
e', 'enabled', 'enables', 'enabling', 'enb', 'enclosed', 'enclosure', 'encoding', 'encounter', 'encount
urage', 'encouraged', 'encourages', 'encouraging', 'encryption', 'encyclopedia', 'end', 'endangered', '
if', 'ending', 'endless', 'endorsed', 'endorsement', 'ends', 'enemies', 'enemy', 'energy', 'enforcement
ngage', 'engaged', 'engagement', 'engaging', 'engine', 'engineer', 'engineering', 'engineers', 'engines
, 'english', 'enhance', 'enhanced', 'enhancement', 'enhancements', 'enhancing', 'enjoy', 'enjoyed', 'en
large', 'enlargement', 'enormous', 'enough', 'enquiries', 'enquiry', 'enrolled', 'enrollment', 'ensembl
, 'ensures', 'ensuring', 'ent', 'enter', 'entered', 'entering', 'enterprise', 'enterprises', 'enters',
g', 'entertainment', 'entire', 'entirely', 'entities', 'entitled', 'entity', 'entrance', 'entrepreneur'
eurs', 'entries', 'entry', 'envelope', 'environment', 'environmental', 'environments', 'enzyme', 'eos',
, 'epic', 'epinions', 'epinionscom', 'episode', 'episodes', 'epson', 'eq', 'equal', 'equality', 'equall
n', 'equations', 'equilibrium', 'equipment', 'equipped', 'equity', 'equivalent', 'er', 'era', 'eric', '
'erik', 'erotic', 'erotica', 'erp', 'error', 'errors', 'es', 'escape', 'escort', 'escorts', 'especially
'essay', 'essays', 'essence', 'essential', 'essentially', 'essentials', 'essex', 'est', 'establish', 'e
'establishing', 'establishment', 'estate', 'estates', 'estimate', 'estimated', 'estimates', 'estimation
, 'et', 'etc', 'eternal', 'ethernet', 'ethical', 'ethics', 'ethiopia', 'ethnic', 'eu', 'eugene', 'eur',
rope', 'european', 'euros', 'ev', 'eva', 'eval', 'evaluate', 'evaluated', 'evaluating', 'evaluation', '
, 'evanescence', 'evans', 'eve', 'even', 'evening', 'event', 'events', 'eventually', 'ever', 'every', '
'everyday', 'everyone', 'everything', 'everywhere', 'evidence', 'evident', 'evil', 'evolution', 'ex', '
ctly', 'exam', 'examination', 'examinations', 'examine', 'examined', 'examines', 'examining', 'example'
, 'exams', 'exceed', 'excel', 'excellence', 'excellent', 'except', 'exception', 'exceptional', 'excepti
pt', 'excess', 'excessive', 'exchange', 'exchanges', 'excited', 'excitement', 'exciting', 'exclude', 'e
xcluding', 'exclusion', 'exclusive', 'exclusively', 'excuse', 'exec', 'execute', 'executed', 'execution
e', 'executives', 'exempt', 'exemption', 'exercise', 'exercises', 'exhaust', 'exhibit', 'exhibition', '
, 'exhibits', 'exist', 'existed', 'existence', 'existing', 'exists', 'exit', 'exotic', 'exp', 'expand',
'expanding', 'expansion', 'expansys', 'expect', 'expectations', 'expected', 'expects', 'expedia', 'expe
xpenditures', 'expense', 'expenses', 'expensive', 'experience', 'experienced', 'experiences', 'experien
riment', 'experimental', 'experiments', 'expert', 'expertise', 'experts', 'expiration', 'expired', 'exp
ain', 'explained', 'explaining', 'explains', 'explanation', 'explicit', 'explicitly', 'exploration', 'e
plorer', 'exploring', 'explosion', 'expo', 'export', 'exports', 'exposed', 'exposure', 'express', 'expr
ression', 'expressions', 'ext', 'extend', 'extended', 'extending', 'extends', 'extension', 'extensions'
e', 'extent', 'exterior', 'external', 'extra', 'extract', 'extraction', 'extraordinary', 'extras', 'ext
emely', 'eye', 'eyed', 'eyes', 'ez', 'f', 'fa', 'fabric', 'fabrics', 'fabulous', 'face', 'faced', 'face
, 'facilitate', 'facilities', 'facility', 'facing', 'fact', 'factor', 'factors', 'factory', 'facts', 'f
il', 'failed', 'failing', 'fails', 'failure', 'failures', 'fair', 'fairfield', 'fairly', 'fairy', 'fait
'fall', 'fallen', 'falling', 'falls', 'false', 'fame', 'familiar', 'families', 'family', 'famous', 'fan
'fans', 'fantastic', 'fantasy', 'faq', 'faqs', 'far', 'fare', 'fares', 'farm', 'farmer', 'farmers', 'fa
ms', 'fascinating', 'fashion', 'fast', 'faster', 'fastest', 'fat', 'fatal', 'fate', 'father', 'fathers'
'fault', 'favor', 'favorite', 'favorites', 'favors', 'favour', 'favourite', 'favourites', 'fax', 'fbi',
, 'fd', 'fda', 'fe', 'fear', 'fears', 'feat', 'feature', 'featured', 'features', 'featuring', 'feb', 'f
ed', 'federal', 'federation', 'fee', 'feed', 'feedback', 'feeding', 'feeds', 'feel', 'feeling', 'feelin
, 'fees', 'feet', 'fell', 'fellow', 'fellowship', 'felt', 'female', 'females', 'fence', 'feof', 'ferrar
'festival', 'festivals', 'fetish', 'fever', 'few', 'fewer', 'ff', 'fg', 'fi', 'fiber', 'fibre', 'fictio
'fields', 'fifteen', 'fifth', 'fifty', 'fig', 'fight', 'fighter', 'fighters', 'fighting', 'figure', 'fi
ures', 'fiji', 'file', 'filed', 'filename', 'files', 'filing', 'fill', 'filled', 'filling', 'film', 'fi
s', 'filter', 'filtering', 'filters', 'fin', 'final', 'finally', 'finals', 'finance', 'finances', 'fina
ancing', 'find', 'findarticles', 'finder', 'finding', 'findings', 'findlaw', 'finds', 'fine', 'finest',
'fingering', 'fingers', 'finish', 'finished', 'finishing', 'finite', 'finland', 'finnish', 'fioricet',
ed', 'firefox', 'fireplace', 'fires', 'firewall', 'firewire', 'firm', 'firms', 'firmware', 'first', 'fi
h', 'fisher', 'fisheries', 'fishing', 'fist', 'fisting', 'fit', 'fitness', 'fits', 'fitted', 'fitting',
x', 'fixed', 'fixes', 'fixtures', 'fl', 'fla', 'flag', 'flags', 'flame', 'flash', 'flashers', 'flashing
'flavor', 'fleece', 'fleet', 'flesh', 'flex', 'flexibility', 'flexible', 'flickr', 'flight', 'flights',
oat', 'floating', 'flood', 'floor', 'flooring', 'floors', 'floppy', 'floral', 'florence', 'florida', 'f
orists', 'flour', 'flow', 'flower', 'flowers', 'flows', 'floyd', 'flu', 'fluid', 'flush', 'flux', 'fly'
'flying', 'fm', 'fo', 'foam', 'focal', 'focus', 'focused', 'focuses', 'focusing', 'fog', 'fold', 'folde
s', 'folding', 'folk', 'folks', 'follow', 'followed', 'following', 'follows', 'font', 'fonts', 'foo', '
s', 'fool', 'foot', 'footage', 'football', 'footwear', 'for', 'forbes', 'forbidden', 'force', 'forced',
'ford', 'forecast', 'forecasts', 'foreign', 'forest', 'forestry', 'forests', 'forever', 'forge', 'forge
, 'forgotten', 'fork', 'form', 'formal', 'format', 'formation', 'formats', 'formatting', 'formed', 'for
rly', 'forming', 'forms', 'formula', 'fort', 'forth', 'fortune', 'forty', 'forum', 'forums', 'forward',
g', 'fossil', 'foster', 'foto', 'fotos', 'fought', 'foul', 'found', 'foundation', 'foundations', 'found
r', 'fountain', 'four', 'fourth', 'fox', 'fp', 'fr', 'fraction', 'fragrance', 'fragrances', 'frame', 'f
mes', 'framework', 'framing', 'france', 'franchise', 'francis', 'francisco', 'frank', 'frankfurt', 'fra
ser', 'fraud', 'fred', 'frederick', 'free', 'freebsd', 'freedom', 'freelance', 'freely', 'freeware', 'f
ight', 'french', 'frequencies', 'frequency', 'frequent', 'frequently', 'fresh', 'fri', 'friday', 'fridg
, 'friendly', 'friends', 'friendship', 'frog', 'from', 'front', 'frontier', 'frontpage', 'frost', 'froz
, 'fruits', 'fs', 'ft', 'ftp', 'fu', 'fuck', 'fucked', 'fucking', 'fuel', 'fuji', 'fujitsu', 'full', 'f
, 'function', 'functional', 'functionality', 'functioning', 'functions', 'fund', 'fundamental', 'fundam
nded', 'funding', 'fundraising', 'funds', 'funeral', 'funk', 'funky', 'funny', 'fur', 'furnished', 'fur
'furniture', 'further', 'furthermore', 'fusion', 'future', 'futures', 'fuzzy', 'fw', 'fwd', 'fx', 'fy',
'gabriel', 'gadgets', 'gage', 'gain', 'gained', 'gains', 'galaxy', 'gale', 'galleries', 'gallery', 'gam
e', 'gamecube', 'games', 'gamespot', 'gaming', 'gamma', 'gang', 'gangbang', 'gap', 'gaps', 'garage', 'g
rcia', 'garden', 'gardening', 'gardens', 'garlic', 'garmin', 'gary', 'gas', 'gasoline', 'gate', 'gates'
'gather', 'gathered', 'gathering', 'gauge', 'gave', 'gay', 'gays', 'gazette', 'gb', 'gba', 'gbp', 'gc',
, 'gdp', 'ge', 'gear', 'geek', 'gel', 'gem', 'gen', 'gender', 'gene', 'genealogy', 'general', 'generall
e', 'generated', 'generates', 'generating', 'generation', 'generations', 'generator', 'generators', 'ge
erous', 'genes', 'genesis', 'genetic', 'genetics', 'geneva', 'genius', 'genome', 'genre', 'genres', 'ge
leman', 'gently', 'genuine', 'geo', 'geographic', 'geographical', 'geography', 'geological', 'geology',
'george', 'georgia', 'gerald', 'german', 'germany', 'get', 'gets', 'getting', 'gg', 'ghana', 'ghost', '
'giant', 'giants', 'gibraltar', 'gibson', 'gif', 'gift', 'gifts', 'gig', 'gilbert', 'girl', 'girlfriend
'gis', 'give', 'given', 'gives', 'giving', 'gl', 'glad', 'glance', 'glasgow', 'glass', 'glasses', 'glen
'global', 'globe', 'glory', 'glossary', 'gloves', 'glow', 'glucose', 'gm', 'gmbh', 'gmc', 'gmt', 'gnome
o', 'goal', 'goals', 'goat', 'god', 'gods', 'goes', 'going', 'gold', 'golden', 'golf', 'gone', 'gonna',
ods', 'google', 'gordon', 'gore', 'gorgeous', 'gospel', 'gossip', 'got', 'gothic', 'goto', 'gotta', 'go
met', 'gov', 'governance', 'governing', 'government', 'governmental', 'governments', 'governor', 'govt'
l', 'gps', 'gr', 'grab', 'grace', 'grad', 'grade', 'grades', 'gradually', 'graduate', 'graduated', 'gra
aduation', 'graham', 'grain', 'grammar', 'grams', 'grand', 'grande', 'granny', 'grant', 'granted', 'gra
h', 'graphic', 'graphical', 'graphics', 'graphs', 'gras', 'grass', 'grateful', 'gratis', 'gratuit', 'gr
ty', 'gray', 'great', 'greater', 'greatest', 'greatly', 'greece', 'greek', 'green', 'greene', 'greenhou
boro', 'greeting', 'greetings', 'greg', 'gregory', 'grenada', 'grew', 'grey', 'grid', 'griffin', 'grill
'grocery', 'groove', 'gross', 'ground', 'grounds', 'groundwater', 'group', 'groups', 'grove', 'grow', '
rown', 'grows', 'growth', 'gs', 'gsm', 'gst', 'gt', 'gtk', 'guam', 'guarantee', 'guaranteed', 'guarante
, 'guardian', 'guards', 'guatemala', 'guess', 'guest', 'guestbook', 'guests', 'gui', 'guidance', 'guide
'guidelines', 'guides', 'guild', 'guilty', 'guinea', 'guitar', 'guitars', 'gulf', 'gun', 'guns', 'guru'
yana', 'guys', 'gym', 'gzip', 'h', 'ha', 'habitat', 'habits', 'hack', 'hacker', 'had', 'hair', 'hairy',
alf', 'halfcom', 'halifax', 'hall', 'halloween', 'halo', 'ham', 'hamburg', 'hamilton', 'hammer', 'hamps
ton', 'hand', 'handbags', 'handbook', 'handed', 'handheld', 'handhelds', 'handjob', 'handjobs', 'handle
, 'handles', 'handling', 'handmade', 'hands', 'handy', 'hang', 'hanging', 'hans', 'hansen', 'happen', '
'happening', 'happens', 'happiness', 'happy', 'harassment', 'harbor', 'harbour', 'hard', 'hardcore', 'h
'harder', 'hardly', 'hardware', 'hardwood', 'harley', 'harm', 'harmful', 'harmony', 'harold', 'harper',
'harrison', 'harry', 'hart', 'hartford', 'harvard', 'harvest', 'harvey', 'has', 'hash', 'hat', 'hate',
e', 'haven', 'having', 'hawaii', 'hawaiian', 'hawk', 'hay', 'hayes', 'hazard', 'hazardous', 'hazards',
'hd', 'hdtv', 'he', 'head', 'headed', 'header', 'headers', 'heading', 'headline', 'headlines', 'headpho
uarters', 'heads', 'headset', 'healing', 'health', 'healthcare', 'healthy', 'hear', 'heard', 'hearing',
'heart', 'hearts', 'heat', 'heated', 'heater', 'heath', 'heather', 'heating', 'heaven', 'heavily', 'hea
w', 'heel', 'height', 'heights', 'held', 'helen', 'helena', 'helicopter', 'hell', 'hello', 'helmet', 'h
d', 'helpful', 'helping', 'helps', 'hence', 'henderson', 'henry', 'hentai', 'hepatitis', 'her', 'herald
'herbal', 'herbs', 'here', 'hereby', 'herein', 'heritage', 'hero', 'heroes', 'herself', 'hewlett', 'hey
i', 'hidden', 'hide', 'hierarchy', 'high', 'higher', 'highest', 'highland', 'highlight', 'highlighted',
s', 'highly', 'highs', 'highway', 'highways', 'hiking', 'hill', 'hills', 'hilton', 'him', 'himself', 'h
t', 'hints', 'hip', 'hire', 'hired', 'hiring', 'his', 'hispanic', 'hist', 'historic', 'historical', 'hi
t', 'hitachi', 'hits', 'hitting', 'hiv', 'hk', 'hl', 'ho', 'hobbies', 'hobby', 'hockey', 'hold', 'holde
, 'holders', 'holding', 'holdings', 'holds', 'hole', 'holes', 'holiday', 'holidays', 'holland', 'hollow
'hollywood', 'holmes', 'holocaust', 'holy', 'home', 'homeland', 'homeless', 'homepage', 'homes', 'homet
ork', 'hon', 'honda', 'honduras', 'honest', 'honey', 'hong', 'honolulu', 'honor', 'honors', 'hood', 'ho
'hope', 'hoped', 'hopefully', 'hopes', 'hoping', 'hopkins', 'horizon', 'horizontal', 'hormone', 'horn',
orrible', 'horror', 'horse', 'horses', 'hose', 'hospital', 'hospitality', 'hospitals', 'host', 'hosted'
'hostels', 'hosting', 'hosts', 'hot', 'hotel', 'hotels', 'hotelscom', 'hotmail', 'hottest', 'hour', 'ho
s', 'house', 'household', 'households', 'houses', 'housewares', 'housewives', 'housing', 'houston', 'ho
, 'however', 'howto', 'hp', 'hq', 'hr', 'href', 'hrs', 'hs', 'ht', 'html', 'http', 'hu', 'hub', 'hudson
'hugh', 'hughes', 'hugo', 'hull', 'human', 'humanitarian', 'humanities', 'humanity', 'humans', 'humidit
'hundred', 'hundreds', 'hung', 'hungarian', 'hungary', 'hunger', 'hungry', 'hunt', 'hunter', 'hunting',
n', 'hurricane', 'hurt', 'husband', 'hwy', 'hybrid', 'hydraulic', 'hydrocodone', 'hydrogen', 'hygiene',
s', 'hypothetical', 'hyundai', 'hz', 'i', 'ia', 'ian', 'ibm', 'ic', 'ice', 'iceland', 'icon', 'icons',
, 'id', 'idaho', 'ide', 'idea', 'ideal', 'ideas', 'identical', 'identification', 'identified', 'identif
ifies', 'identify', 'identifying', 'identity', 'idle', 'idol', 'ids', 'ie', 'ieee', 'if', 'ignore', 'ig
, 'iii', 'il', 'ill', 'illegal', 'illinois', 'illness', 'illustrated', 'illustration', 'illustrations',
, 'image', 'images', 'imagination', 'imagine', 'imaging', 'img', 'immediate', 'immediately', 'immigrant
tion', 'immune', 'immunology', 'impact', 'impacts', 'impaired', 'imperial', 'implement', 'implementatio
nted', 'implementing', 'implications', 'implied', 'implies', 'import', 'importance', 'important', 'impo
mported', 'imports', 'impose', 'imposed', 'impossible', 'impressed', 'impression', 'impressive', 'impro
ed', 'improvement', 'improvements', 'improving', 'in', 'inappropriate', 'inbox', 'inc', 'incentive', 'i
'incest', 'inch', 'inches', 'incidence', 'incident', 'incidents', 'incl', 'include', 'included', 'inclu
ding', 'inclusion', 'inclusive', 'income', 'incoming', 'incomplete', 'incorporate', 'incorporated', 'in
ncrease', 'increased', 'increases', 'increasing', 'increasingly', 'incredible', 'incurred', 'ind', 'ind
endence', 'independent', 'independently', 'index', 'indexed', 'indexes', 'india', 'indian', 'indiana',
s', 'indians', 'indicate', 'indicated', 'indicates', 'indicating', 'indication', 'indicator', 'indicato
s', 'indie', 'indigenous', 'indirect', 'individual', 'individually', 'individuals', 'indonesia', 'indon
oor', 'induced', 'induction', 'industrial', 'industries', 'industry', 'inexpensive', 'inf', 'infant', '
nfected', 'infection', 'infections', 'infectious', 'infinite', 'inflation', 'influence', 'influenced',
, 'info', 'inform', 'informal', 'information', 'informational', 'informative', 'informed', 'infrared',
ure', 'ing', 'ingredients', 'inherited', 'initial', 'initially', 'initiated', 'initiative', 'initiative
on', 'injured', 'injuries', 'injury', 'ink', 'inkjet', 'inline', 'inn', 'inner', 'innocent', 'innovatio
ions', 'innovative', 'inns', 'input', 'inputs', 'inquire', 'inquiries', 'inquiry', 'ins', 'insects', 'i
erted', 'insertion', 'inside', 'insider', 'insight', 'insights', 'inspection', 'inspections', 'inspecto
tion', 'inspired', 'install', 'installation', 'installations', 'installed', 'installing', 'instance', '
'instant', 'instantly', 'instead', 'institute', 'institutes', 'institution', 'institutional', 'institut
ruction', 'instructional', 'instructions', 'instructor', 'instructors', 'instrument', 'instrumental', '
ion', 'instruments', 'insulin', 'insurance', 'insured', 'int', 'intake', 'integer', 'integral', 'integr
rated', 'integrating', 'integration', 'integrity', 'intel', 'intellectual', 'intelligence', 'intelligen
, 'intended', 'intense', 'intensity', 'intensive', 'intent', 'intention', 'inter', 'interact', 'interac
ractions', 'interactive', 'interest', 'interested', 'interesting', 'interests', 'interface', 'interface
rence', 'interim', 'interior', 'intermediate', 'internal', 'international', 'internationally', 'interne
hip', 'interpretation', 'interpreted', 'interracial', 'intersection', 'interstate', 'interval', 'interv
vention', 'interventions', 'interview', 'interviews', 'intimate', 'intl', 'into', 'intranet', 'intro',
'introduced', 'introduces', 'introducing', 'introduction', 'introductory', 'invalid', 'invasion', 'inve
entory', 'invest', 'investigate', 'investigated', 'investigation', 'investigations', 'investigator', 'i
s', 'investing', 'investment', 'investments', 'investor', 'investors', 'invisible', 'invision', 'invita
tations', 'invite', 'invited', 'invoice', 'involve', 'involved', 'involvement', 'involves', 'involving'
```

```
n', 'iowa', 'ip', 'ipaq', 'ipod', 'ips', 'ir', 'ira', 'iran', 'iraq', 'iraqi', 'irc', 'ireland', 'irish
'irrigation', 'irs', 'is', 'isa', 'isaac', 'isbn', 'islam', 'islamic', 'island', 'islands', 'isle', 'is
d', 'isolation', 'isp', 'israel', 'israeli', 'issn', 'issue', 'issued', 'issues', 'ist', 'istanbul', 'i
, 'italian', 'italiano', 'italic', 'italy', 'item', 'items', 'its', 'itsa', 'itself', 'itunes', 'iv', '
, 'j', 'ja', 'jack', 'jacket', 'jackets', 'jackie', 'jackson', 'jacksonville', 'jacob', 'jade', 'jaguar
'jake', 'jam', 'jamaica', 'james', 'jamie', 'jan', 'jane', 'janet', 'january', 'japan', 'japanese', 'ja
'java', 'javascript', 'jay', 'jazz', 'jc', 'jd', 'je', 'jean', 'jeans', 'jeep', 'jeff', 'jefferson', 'j
lsoft', 'jennifer', 'jenny', 'jeremy', 'jerry', 'jersey', 'jerusalem', 'jesse', 'jessica', 'jesus', 'je
'jewel', 'jewellery', 'jewelry', 'jewish', 'jews', 'jill', 'jim', 'jimmy', 'jj', 'jm', 'jo', 'joan', 'j
'joe', 'joel', 'john', 'johnny', 'johns', 'johnson', 'johnston', 'join', 'joined', 'joining', 'joins', 
ke', 'jokes', 'jon', 'jonathan', 'jones', 'jordan', 'jose', 'joseph', 'josh', 'joshua', 'journal', 'jou
ournalist', 'journalists', 'journals', 'journey', 'joy', 'joyce', 'jp', 'jpeg', 'jpg', 'jr', 'js', 'jua
'judges', 'judgment', 'judicial', 'judy', 'juice', 'jul', 'julia', 'julian', 'julie', 'july', 'jump', '
un', 'junction', 'june', 'jungle', 'junior', 'junk', 'jurisdiction', 'jury', 'just', 'justice', 'justif
, 'juvenile', 'jvc', 'k', 'ka', 'kai', 'kansas', 'karaoke', 'karen', 'karl', 'karma', 'kate', 'kathy', 
trina', 'kay', 'kazakhstan', 'kb', 'kde', 'keen', 'keep', 'keeping', 'keeps', 'keith', 'kelkoo', 'kelly
ennedy', 'kenneth', 'kenny', 'keno', 'kent', 'kentucky', 'kenya', 'kept', 'kernel', 'kerry', 'kevin', '
ard', 'keyboards', 'keys', 'keyword', 'keywords', 'kg', 'kick', 'kid', 'kidney', 'kids', 'kijiji', 'kil
, 'killer', 'killing', 'kills', 'kilometers', 'kim', 'kinase', 'kind', 'kinda', 'kinds', 'king', 'kingd
, 'kingston', 'kirk', 'kiss', 'kissing', 'kit', 'kitchen', 'kits', 'kitty', 'klein', 'km', 'knee', 'kne
'knight', 'knights', 'knit', 'knitting', 'knives', 'knock', 'know', 'knowing', 'knowledge', 'knowledges
n', 'knows', 'ko', 'kodak', 'kong', 'korea', 'korean', 'kruger', 'ks', 'kurt', 'kuwait', 'kw', 'ky', 'k
'la', 'lab', 'label', 'labeled', 'labels', 'labor', 'laboratories', 'laboratory', 'labour', 'labs', 'la
'ladder', 'laden', 'ladies', 'lady', 'lafayette', 'laid', 'lake', 'lakes', 'lamb', 'lambda', 'lamp', 'l
, 'lancaster', 'lance', 'land', 'landing', 'lands', 'landscape', 'landscapes', 'lane', 'lanes', 'lang',
'languages', 'lanka', 'lap', 'laptop', 'laptops', 'large', 'largely', 'larger', 'largest', 'larry', 'la
'last', 'lasting', 'lat', 'late', 'lately', 'later', 'latest', 'latex', 'latin', 'latina', 'latinas', '
titude', 'latter', 'latvia', 'lauderdale', 'laugh', 'laughing', 'launch', 'launched', 'launches', 'laun
a', 'lauren', 'law', 'lawn', 'lawrence', 'laws', 'lawsuit', 'lawyer', 'lawyers', 'lay', 'layer', 'layer
, 'lazy', 'lb', 'lbs', 'lc', 'lcd', 'ld', 'le', 'lead', 'leader', 'leaders', 'leadership', 'leading', '
f', 'league', 'lean', 'learn', 'learned', 'learners', 'learning', 'lease', 'leasing', 'least', 'leather
'leaves', 'leaving', 'lebanon', 'lecture', 'lectures', 'led', 'lee', 'leeds', 'left', 'leg', 'legacy', 
gally', 'legend', 'legendary', 'legends', 'legislation', 'legislative', 'legislature', 'legitimate', 'l
re', 'lemon', 'len', 'lender', 'lenders', 'lending', 'length', 'lens', 'lenses', 'leo', 'leon', 'leonar
'les', 'lesbian', 'lesbians', 'leslie', 'less', 'lesser', 'lesson', 'lessons', 'let', 'lets', 'letter',
'letting', 'leu', 'level', 'levels', 'levitra', 'levy', 'lewis', 'lexington', 'lexmark', 'lexus', 'lf',
'liabilities', 'liability', 'liable', 'lib', 'liberal', 'liberia', 'liberty', 'librarian', 'libraries',
'libs', 'licence', 'license', 'licensed', 'licenses', 'licensing', 'licking', 'lid', 'lie', 'liechtenst
, 'life', 'lifestyle', 'lifetime', 'lift', 'light', 'lighter', 'lighting', 'lightning', 'lights', 'ligh
ike', 'liked', 'likelihood', 'likely', 'likes', 'likewise', 'lil', 'lime', 'limit', 'limitation', 'limi
imited', 'limiting', 'limits', 'limousines', 'lincoln', 'linda', 'lindsay', 'line', 'linear', 'lined', 
ngerie', 'link', 'linked', 'linking', 'links', 'linux', 'lion', 'lions', 'lip', 'lips', 'liquid', 'lisa
'listed', 'listen', 'listening', 'listing', 'listings', 'listprice', 'lists', 'lit', 'lite', 'literacy'
y', 'literary', 'literature', 'lithuania', 'litigation', 'little', 'live', 'livecam', 'lived', 'liver',
, 'lives', 'livesex', 'livestock', 'living', 'liz', 'll', 'llc', 'lloyd', 'llp', 'lm', 'ln', 'lo', 'loa
, 'loading', 'loads', 'loan', 'loans', 'lobby', 'loc', 'local', 'locale', 'locally', 'locate', 'located
n', 'locations', 'locator', 'lock', 'locked', 'locking', 'locks', 'lodge', 'lodging', 'log', 'logan', '
gging', 'logic', 'logical', 'login', 'logistics', 'logitech', 'logo', 'logos', 'logs', 'lol', 'lolita',
'lone', 'lonely', 'long', 'longer', 'longest', 'longitude', 'look', 'looked', 'looking', 'looks', 'look
kup', 'loop', 'loops', 'loose', 'lopez', 'lord', 'los', 'lose', 'losing', 'loss', 'losses', 'lost', 'lo
'lottery', 'lotus', 'lou', 'loud', 'louis', 'louise', 'louisiana', 'louisville', 'lounge', 'love', 'lov
y', 'lover', 'lovers', 'loves', 'loving', 'low', 'lower', 'lowest', 'lows', 'lp', 'ls', 'lt', 'ltd', 'l
'lucia', 'luck', 'lucky', 'lucy', 'luggage', 'luis', 'luke', 'lunch', 'lung', 'luther', 'luxembourg', '
cos', 'lying', 'lynn', 'lyric', 'lyrics', 'm', 'ma', 'mac', 'macedonia', 'machine', 'machinery', 'machi
tosh', 'macro', 'macromedia', 'mad', 'madagascar', 'made', 'madison', 'madness', 'madonna', 'madrid', '
'magazine', 'magazines', 'magic', 'magical', 'magnet', 'magnetic', 'magnificent', 'magnitude', 'mai', '
il', 'mailed', 'mailing', 'mailman', 'mails', 'mailto', 'main', 'maine', 'mainland', 'mainly', 'mainstr
ain', 'maintained', 'maintaining', 'maintains', 'maintenance', 'major', 'majority', 'make', 'maker', 'm
es', 'makeup', 'making', 'malawi', 'malaysia', 'maldives', 'male', 'males', 'mali', 'mall', 'malpractic
'mambo', 'man', 'manage', 'managed', 'management', 'manager', 'managers', 'managing', 'manchester', 'ma
datory', 'manga', 'manhattan', 'manitoba', 'manner', 'manor', 'manual', 'manually', 'manuals', 'manufac
factured', 'manufacturer', 'manufacturers', 'manufacturing', 'many', 'map', 'maple', 'mapping', 'maps',
athon', 'marble', 'marc', 'march', 'marco', 'marcus', 'mardi', 'margaret', 'margin', 'maria', 'mariah',
arijuana', 'marilyn', 'marina', 'marine', 'mario', 'marion', 'maritime', 'mark', 'marked', 'marker', 'm
rket', 'marketing', 'marketplace', 'markets', 'marking', 'marks', 'marriage', 'married', 'marriott', 'm
all', 'mart', 'martha', 'martial', 'martin', 'marvel', 'mary', 'maryland', 'mas', 'mask', 'mason', 'mas
usetts', 'massage', 'massive', 'master', 'mastercard', 'masters', 'masturbating', 'masturbation', 'mat'
'matched', 'matches', 'matching', 'mate', 'material', 'materials', 'maternity', 'math', 'mathematical',
s', 'mating', 'matrix', 'mats', 'matt', 'matter', 'matters', 'matthew', 'mattress', 'mature', 'maui', '
'max', 'maximize', 'maximum', 'may', 'maybe', 'mayor', 'mazda', 'mb', 'mba', 'mc', 'mcdonald', 'md', 'm
'meals', 'mean', 'meaning', 'meaningful', 'means', 'meant', 'meanwhile', 'measure', 'measured', 'measur
surements', 'measures', 'measuring', 'meat', 'mechanical', 'mechanics', 'mechanism', 'mechanisms', 'med
'media', 'median', 'medicaid', 'medical', 'medicare', 'medication', 'medications', 'medicine', 'medicin
al', 'meditation', 'mediterranean', 'medium', 'medline', 'meet', 'meeting', 'meetings', 'meets', 'meetu
'mel', 'melbourne', 'melissa', 'mem', 'member', 'members', 'membership', 'membrane', 'memo', 'memorabil
al', 'memories', 'memory', 'memphis', 'men', 'mens', 'ment', 'mental', 'mention', 'mentioned', 'mentor'
enus', 'mercedes', 'merchandise', 'merchant', 'merchants', 'mercury', 'mercy', 'mere', 'merely', 'merge
'merit', 'merry', 'mesa', 'mesh', 'mess', 'message', 'messages', 'messaging', 'messenger', 'met', 'meta
sm', 'metadata', 'metal', 'metallic', 'metallica', 'metals', 'meter', 'meters', 'method', 'methodology'
'metres', 'metric', 'metro', 'metropolitan', 'mexican', 'mexico', 'meyer', 'mf', 'mfg', 'mg', 'mh', 'mh
ia', 'miami', 'mic', 'mice', 'michael', 'michel', 'michelle', 'michigan', 'micro', 'microphone', 'micro
owave', 'mid', 'middle', 'midi', 'midlands', 'midnight', 'midwest', 'might', 'mighty', 'migration', 'mi
'milan', 'mild', 'mile', 'mileage', 'miles', 'milf', 'milfhunter', 'milfs', 'military', 'milk', 'mill',
m', 'miller', 'million', 'millions', 'mills', 'milton', 'milwaukee', 'mime', 'min', 'mind', 'minds', 'm
al', 'minerals', 'mines', 'mini', 'miniature', 'minimal', 'minimize', 'minimum', 'mining', 'minister', 
'ministries', 'ministry', 'minneapolis', 'minnesota', 'minolta', 'minor', 'minority', 'mins', 'mint', '
ute', 'minutes', 'miracle', 'mirror', 'mirrors', 'misc', 'miscellaneous', 'miss', 'missed', 'missile', 
'mission', 'missions', 'mississippi', 'missouri', 'mistake', 'mistakes', 'mistress', 'mit', 'mitchell',
i', 'mix', 'mixed', 'mixer', 'mixing', 'mixture', 'mj', 'ml', 'mlb', 'mls', 'mm', 'mn', 'mo', 'mobile',
'mobility', 'mod', 'mode', 'model', 'modeling', 'modelling', 'models', 'modem', 'modems', 'moderate', '
'moderators', 'modern', 'modes', 'modification', 'modifications', 'modified', 'modify', 'mods', 'modula
, 'modules', 'moisture', 'mold', 'moldova', 'molecular', 'molecules', 'mom', 'moment', 'moments', 'mome
s', 'mon', 'monaco', 'monday', 'monetary', 'money', 'mongolia', 'monica', 'monitor', 'monitored', 'moni
nitors', 'monkey', 'mono', 'monroe', 'monster', 'montana', 'monte', 'montgomery', 'month', 'monthly', '
ntreal', 'mood', 'moon', 'moore', 'moral', 'more', 'moreover', 'morgan', 'morning', 'morocco', 'morris'
, 'mortality', 'mortgage', 'mortgages', 'moscow', 'moses', 'moss', 'most', 'mostly', 'motel', 'motels',
'motherboard', 'mothers', 'motion', 'motivated', 'motivation', 'motor', 'motorcycle', 'motorcycles', 'm
otors', 'mount', 'mountain', 'mountains', 'mounted', 'mounting', 'mounts', 'mouse', 'mouth', 'move', 'm
ment', 'movements', 'movers', 'moves', 'movie', 'movies', 'moving', 'mozambique', 'mozilla', 'mp', 'mpe
'mpg', 'mph', 'mr', 'mrna', 'mrs', 'ms', 'msg', 'msgid', 'msgstr', 'msie', 'msn', 'mt', 'mtv', 'mu', 'm
'mug', 'multi', 'multimedia', 'multiple', 'mumbai', 'munich', 'municipal', 'municipality', 'murder', 'm
ray', 'muscle', 'muscles', 'museum', 'museums', 'music', 'musical', 'musician', 'musicians', 'muslim',
'must', 'mustang', 'mutual', 'muze', 'mv', 'mw', 'mx', 'my', 'myanmar', 'myers', 'myrtle', 'myself', 'm
space', 'mysql', 'mysterious', 'mystery', 'myth', 'n', 'na', 'nail', 'nails', 'naked', 'nam', 'name', '
ely', 'names', 'namespace', 'namibia', 'nancy', 'nano', 'naples', 'narrative', 'narrow', 'nasa', 'nasca
, 'nashville', 'nasty', 'nat', 'nathan', 'nation', 'national', 'nationally', 'nations', 'nationwide', '
to', 'natural', 'naturally', 'naturals', 'nature', 'naughty', 'nav', 'naval', 'navigate', 'navigation',
, 'navy', 'nb', 'nba', 'nbc', 'nc', 'ncaa', 'nd', 'ne', 'near', 'nearby', 'nearest', 'nearly', 'nebrask
'necessarily', 'necessary', 'necessity', 'neck', 'necklace', 'need', 'needed', 'needle', 'needs', 'nega
tiation', 'negotiations', 'neighbor', 'neighborhood', 'neighbors', 'neil', 'neither', 'nelson', 'neo',
al', 'nerve', 'nervous', 'nest', 'nested', 'net', 'netherlands', 'netscape', 'network', 'networking', '
'neural', 'neutral', 'nevada', 'never', 'nevertheless', 'new', 'newark', 'newbie', 'newcastle', 'newer'
'newfoundland', 'newly', 'newport', 'news', 'newscom', 'newsletter', 'newsletters', 'newspaper', 'newsp
ton', 'next', 'nextel', 'nfl', 'ng', 'nh', 'nhl', 'nhs', 'ni', 'niagara', 'nicaragua', 'nice', 'nichola
'nickel', 'nickname', 'nicole', 'niger', 'nigeria', 'night', 'nightlife', 'nightmare', 'nights', 'nike'
'nil', 'nine', 'nintendo', 'nipple', 'nipples', 'nirvana', 'nissan', 'nitrogen', 'nj', 'nl', 'nm', 'nn'
le', 'nobody', 'node', 'nodes', 'noise', 'nokia', 'nominated', 'nomination', 'nominations', 'non', 'non
it', 'noon', 'nor', 'norfolk', 'norm', 'normal', 'normally', 'norman', 'north', 'northeast', 'northern'
t', 'norton', 'norway', 'norwegian', 'nos', 'nose', 'not', 'note', 'notebook', 'notebooks', 'noted', 'n
ing', 'notice', 'noticed', 'notices', 'notification', 'notifications', 'notified', 'notify', 'notion',
ttingham', 'nov', 'nova', 'novel', 'novels', 'novelty', 'november', 'now', 'nowhere', 'np', 'nr', 'ns',
, 'ntsc', 'nu', 'nuclear', 'nude', 'nudist', 'nudity', 'nuke', 'null', 'number', 'numbers', 'numeric',
'numerous', 'nurse', 'nursery', 'nurses', 'nursing', 'nut', 'nutrition', 'nutritional', 'nuts', 'nutten
idia', 'nw', 'ny', 'nyc', 'nylon', 'nz', 'o', 'oak', 'oakland', 'oaks', 'oasis', 'ob', 'obesity', 'obit
j', 'object', 'objective', 'objectives', 'objects', 'obligation', 'obligations', 'observation', 'observ
serve', 'observed', 'observer', 'obtain', 'obtained', 'obtaining', 'obvious', 'obviously', 'oc', 'occas
ional', 'occasionally', 'occasions', 'occupation', 'occupational', 'occupations', 'occupied', 'occur',
'occurrence', 'occurring', 'occurs', 'ocean', 'oclc', 'oct', 'october', 'odd', 'odds', 'oe', 'oecd', 'o
'off', 'offense', 'offensive', 'offer', 'offered', 'offering', 'offerings', 'offers', 'office', 'office
s', 'offices', 'official', 'officially', 'officials', 'offline', 'offset', 'offshore', 'often', 'og', '
'oil', 'oils', 'ok', 'okay', 'oklahoma', 'ol', 'old', 'older', 'oldest', 'olive', 'oliver', 'olympic',
'olympus', 'om', 'omaha', 'oman', 'omega', 'omissions', 'on', 'once', 'one', 'ones', 'ongoing', 'onion'
'only', 'ons', 'ontario', 'onto', 'oo', 'ooo', 'oops', 'op', 'open', 'opened', 'opening', 'openings', '
ra', 'operate', 'operated', 'operates', 'operating', 'operation', 'operational', 'operations', 'operato
rs', 'opinion', 'opinions', 'opponent', 'opponents', 'opportunities', 'opportunity', 'opposed', 'opposi
tion', 'opt', 'optical', 'optics', 'optimal', 'optimization', 'optimize', 'optimum', 'option', 'optiona
s', 'or', 'oracle', 'oral', 'orange', 'orbit', 'orchestra', 'order', 'ordered', 'ordering', 'orders', '
'ordinary', 'oregon', 'org', 'organ', 'organic', 'organisation', 'organisations', 'organised', 'organis
zation', 'organizational', 'organizations', 'organize', 'organized', 'organizer', 'organizing', 'orgasm
'oriental', 'orientation', 'oriented', 'origin', 'original', 'originally', 'origins', 'orlando', 'orlea
'oscar', 'ot', 'other', 'others', 'otherwise', 'ottawa', 'ou', 'ought', 'our', 'ours', 'ourselves', 'ou
e', 'outcomes', 'outdoor', 'outdoors', 'outer', 'outlet', 'outline', 'outlined', 'outlook', 'output', '
utreach', 'outside', 'outsourcing', 'outstanding', 'oval', 'oven', 'over', 'overall', 'overcome', 'over
night', 'overseas', 'overview', 'owen', 'own', 'owned', 'owner', 'owners', 'ownership', 'owns', 'oxford
'oxygen', 'oz', 'ozone', 'p', 'pa', 'pac', 'pace', 'pacific', 'pack', 'package', 'packages', 'packaging
, 'packed', 'packet', 'packets', 'packing', 'packs', 'pad', 'pads', 'page', 'pages', 'paid', 'pain', 'p
int', 'paintball', 'painted', 'painting', 'paintings', 'pair', 'pairs', 'pakistan', 'pal', 'palace', 'p
tine', 'palestinian', 'palm', 'palmer', 'pam', 'pamela', 'pan', 'panama', 'panasonic', 'panel', 'panels
'panties', 'pants', 'pantyhose', 'paper', 'paperback', 'paperbacks', 'papers', 'papua', 'par', 'para',
aradise', 'paragraph', 'paragraphs', 'paraguay', 'parallel', 'parameter', 'parameters', 'parcel', 'pare
al', 'parenting', 'parents', 'paris', 'parish', 'park', 'parker', 'parking', 'parks', 'parliament', 'pa
, 'part', 'partial', 'partially', 'participant', 'participants', 'participate', 'participated', 'partic
articipation', 'particle', 'particles', 'particular', 'particularly', 'parties', 'partition', 'partly',
'partners', 'partnership', 'partnerships', 'parts', 'party', 'pas', 'paso', 'pass', 'passage', 'passed'
r', 'passengers', 'passes', 'passing', 'passion', 'passive', 'passport', 'password', 'passwords', 'past
'paste', 'pastor', 'pat', 'patch', 'patches', 'patent', 'patents', 'path', 'pathology', 'paths', 'patie
ts', 'patio', 'patricia', 'patrick', 'patrol', 'pattern', 'patterns', 'paul', 'pavilion', 'paxil', 'pay
, 'payday', 'paying', 'payment', 'payments', 'paypal', 'payroll', 'pays', 'pb', 'pc', 'pci', 'pcs', 'pc
da', 'pdas', 'pdf', 'pdt', 'pe', 'peace', 'peaceful', 'peak', 'pearl', 'peas', 'pediatric', 'pee', 'pee
, 'peers', 'pen', 'penalties', 'penalty', 'pencil', 'pendant', 'pending', 'penetration', 'penguin', 'pe
enis', 'penn', 'pennsylvania', 'penny', 'pens', 'pension', 'pensions', 'pentium', 'people', 'peoples',
er', 'perceived', 'percent', 'percentage', 'perception', 'perfect', 'perfectly', 'perform', 'performanc
ances', 'performed', 'performer', 'performing', 'performs', 'perfume', 'perhaps', 'period', 'periodic',
ly', 'periods', 'peripheral', 'peripherals', 'perl', 'permalink', 'permanent', 'permission', 'permissio
t', 'permits', 'permitted', 'perry', 'persian', 'persistent', 'person', 'personal', 'personality', 'per
'personally', 'personals', 'personnel', 'persons', 'perspective', 'perspectives', 'perth', 'peru', 'pes
'pete', 'peter', 'petersburg', 'peterson', 'petite', 'petition', 'petroleum', 'pets', 'pf', 'pg', 'pgp'
ntom', 'pharmaceutical', 'pharmaceuticals', 'pharmacies', 'pharmacology', 'pharmacy', 'phase', 'phases'
enomenon', 'phentermine', 'phi', 'phil', 'philadelphia', 'philip', 'philippines', 'philips', 'phillips'
y', 'phoenix', 'phone', 'phones', 'photo', 'photograph', 'photographer', 'photographers', 'photographic
phs', 'photography', 'photos', 'photoshop', 'php', 'phpbb', 'phrase', 'phrases', 'phys', 'physical', 'p
'physician', 'physicians', 'physics', 'physiology', 'pi', 'piano', 'pic', 'pichunter', 'pick', 'picked'
'picks', 'pickup', 'picnic', 'pics', 'picture', 'pictures', 'pie', 'piece', 'pieces', 'pierce', 'pierre
ike', 'pill', 'pillow', 'pills', 'pilot', 'pin', 'pine', 'ping', 'pink', 'pins', 'pioneer', 'pipe', 'pi
pes', 'pirates', 'piss', 'pissing', 'pit', 'pitch', 'pittsburgh', 'pix', 'pixel', 'pixels', 'pizza', 'p
l', 'place', 'placed', 'placement', 'places', 'placing', 'plain', 'plains', 'plaintiff', 'plan', 'plane
'planet', 'planets', 'planned', 'planner', 'planners', 'planning', 'plans', 'plant', 'plants', 'plasma'
'plastics', 'plate', 'plates', 'platform', 'platforms', 'platinum', 'play', 'playback', 'playboy', 'pla
```

```
r', 'players', 'playing', 'playlist', 'plays', 'playstation', 'plaza', 'plc', 'pleasant', 'please', 'pl
asure', 'pledge', 'plenty', 'plot', 'plots', 'plug', 'plugin', 'plugins', 'plumbing', 'plus', 'plymouth
c', 'pmid', 'pn', 'po', 'pocket', 'pockets', 'pod', 'podcast', 'podcasts', 'poem', 'poems', 'poet', 'po
t', 'pointed', 'pointer', 'pointing', 'points', 'pokemon', 'poker', 'poland', 'polar', 'pole', 'police'
, 'policy', 'polish', 'polished', 'political', 'politicians', 'politics', 'poll', 'polls', 'pollution',
ly', 'polyester', 'polymer', 'polyphonic', 'pond', 'pontiac', 'pool', 'pools', 'poor', 'pop', 'pope', '
opularity', 'population', 'populations', 'por', 'porcelain', 'pork', 'porn', 'porno', 'porsche', 'port'
, 'portal', 'porter', 'portfolio', 'portion', 'portions', 'portland', 'portrait', 'portraits', 'ports',
h', 'portugal', 'portuguese', 'pos', 'pose', 'posing', 'position', 'positioning', 'positions', 'positiv
s', 'possession', 'possibilities', 'possibility', 'possible', 'possibly', 'post', 'postage', 'postal',
'postcards', 'posted', 'poster', 'posters', 'posting', 'postings', 'postposted', 'posts', 'pot', 'potat
s', 'potential', 'potentially', 'potter', 'pottery', 'poultry', 'pound', 'pounds', 'pour', 'poverty', '
well', 'power', 'powered', 'powerful', 'powerpoint', 'powers', 'powerseller', 'pp', 'ppc', 'ppm', 'pr',
, 'practice', 'practices', 'practitioner', 'practitioners', 'prague', 'prairie', 'praise', 'pray', 'pra
rs', 'pre', 'preceding', 'precious', 'precipitation', 'precise', 'precisely', 'precision', 'predict', '
'prediction', 'predictions', 'prefer', 'preference', 'preferences', 'preferred', 'prefers', 'prefix', '
'pregnant', 'preliminary', 'premier', 'premiere', 'premises', 'premium', 'prep', 'prepaid', 'preparatio
e', 'prepared', 'preparing', 'prerequisite', 'prescribed', 'prescription', 'presence', 'present', 'pres
'presentations', 'presented', 'presenting', 'presently', 'presents', 'preservation', 'preserve', 'presi
idential', 'press', 'pressed', 'pressing', 'pressure', 'preston', 'pretty', 'prev', 'prevent', 'prevent
ntion', 'preview', 'previews', 'previous', 'previously', 'price', 'priced', 'prices', 'pricing', 'pride
'primarily', 'primary', 'prime', 'prince', 'princess', 'princeton', 'principal', 'principle', 'principl
, 'printable', 'printed', 'printer', 'printers', 'printing', 'prints', 'prior', 'priorities', 'priority
'prisoner', 'prisoners', 'privacy', 'private', 'privilege', 'privileges', 'prix', 'prize', 'prizes', 'p
ility', 'probably', 'probe', 'problem', 'problems', 'proc', 'procedure', 'procedures', 'proceed', 'proc
oceedings', 'proceeds', 'process', 'processed', 'processes', 'processing', 'processor', 'processors', '
, 'produce', 'produced', 'producer', 'producers', 'produces', 'producing', 'product', 'production', 'pr
'productive', 'productivity', 'products', 'prof', 'profession', 'professional', 'professionals', 'profe
ile', 'profiles', 'profit', 'profits', 'program', 'programme', 'programmer', 'programmers', 'programmes
ing', 'programs', 'progress', 'progressive', 'prohibited', 'project', 'projected', 'projection', 'proje
ectors', 'projects', 'prominent', 'promise', 'promised', 'promises', 'promising', 'promo', 'promote', '
'promotes', 'promoting', 'promotion', 'promotional', 'promotions', 'prompt', 'promptly', 'proof', 'prop
er', 'properly', 'properties', 'property', 'prophet', 'proportion', 'proposal', 'proposals', 'propose',
'proposition', 'proprietary', 'pros', 'prospect', 'prospective', 'prospects', 'prostate', 'prostores',
tect', 'protected', 'protecting', 'protection', 'protective', 'protein', 'proteins', 'protest', 'protoc
ols', 'prototype', 'proud', 'proudly', 'prove', 'proved', 'proven', 'provide', 'provided', 'providence'
, 'providers', 'provides', 'providing', 'province', 'provinces', 'provincial', 'provision', 'provisions
'prozac', 'ps', 'psi', 'psp', 'pst', 'psychiatry', 'psychological', 'psychology', 'pt', 'pts', 'pty', '
c', 'publication', 'publications', 'publicity', 'publicly', 'publish', 'published', 'publisher', 'publi
lishing', 'pubmed', 'pubs', 'puerto', 'pull', 'pulled', 'pulling', 'pulse', 'pump', 'pumps', 'punch', '
'punk', 'pupils', 'puppy', 'purchase', 'purchased', 'purchases', 'purchasing', 'pure', 'purple', 'purpo
es', 'purse', 'pursuant', 'pursue', 'pursuit', 'push', 'pushed', 'pushing', 'pussy', 'put', 'puts', 'pu
zle', 'puzzles', 'pvc', 'python', 'q', 'qatar', 'qc', 'qld', 'qt', 'qty', 'quad', 'qualification', 'qua
, 'qualified', 'qualify', 'qualifying', 'qualities', 'quality', 'quantitative', 'quantities', 'quantity
, 'quarter', 'quarterly', 'quarters', 'que', 'quebec', 'queen', 'queens', 'queensland', 'queries', 'que
, 'question', 'questionnaire', 'questions', 'queue', 'qui', 'quick', 'quickly', 'quiet', 'quilt', 'quit
'quiz', 'quizzes', 'quotations', 'quote', 'quoted', 'quotes', 'r', 'ra', 'rabbit', 'race', 'races', 'ra
al', 'racing', 'rack', 'racks', 'radar', 'radiation', 'radical', 'radio', 'radios', 'radius', 'rage', '
l', 'railroad', 'railway', 'rain', 'rainbow', 'raise', 'raised', 'raises', 'raising', 'raleigh', 'rally
'ram', 'ran', 'ranch', 'rand', 'random', 'randy', 'range', 'rangers', 'ranges', 'ranging', 'rank', 'ran
ng', 'rankings', 'ranks', 'rap', 'rape', 'rapid', 'rapidly', 'rapids', 'rare', 'rarely', 'rat', 'rate',
ates', 'rather', 'rating', 'ratings', 'ratio', 'rational', 'ratios', 'rats', 'raw', 'ray', 'raymond', '
'rc', 'rca', 'rd', 're', 'reach', 'reached', 'reaches', 'reaching', 'reaction', 'reactions', 'read', 'r
ders', 'readily', 'reading', 'readings', 'reads', 'ready', 'real', 'realistic', 'reality', 'realize', '
'really', 'realm', 'realtor', 'realtors', 'realty', 'rear', 'reason', 'reasonable', 'reasonably', 'reas
sons', 'rebate', 'rebates', 'rebecca', 'rebel', 'rebound', 'rec', 'recall', 'receipt', 'receive', 'rece
iver', 'receivers', 'receives', 'receiving', 'recent', 'recently', 'reception', 'receptor', 'receptors'
'recipes', 'recipient', 'recipients', 'recognised', 'recognition', 'recognize', 'recognized', 'recommen
ndation', 'recommendations', 'recommended', 'recommends', 'reconstruction', 'record', 'recorded', 'reco
rders', 'recording', 'recordings', 'records', 'recover', 'recovered', 'recovery', 'recreation', 'recrea
cruiting', 'recruitment', 'recycling', 'red', 'redeem', 'redhead', 'reduce', 'reduced', 'reduces', 'red
uction', 'reductions', 'reed', 'reef', 'reel', 'ref', 'refer', 'reference', 'referenced', 'references',
'referrals', 'referred', 'referring', 'refers', 'refinance', 'refine', 'refined', 'reflect', 'reflected
on', 'reflections', 'reflects', 'reform', 'reforms', 'refresh', 'refrigerator', 'refugees', 'refund', '
, 'refuse', 'refused', 'reg', 'regard', 'regarded', 'regarding', 'regardless', 'regards', 'reggae', 're
on', 'regional', 'regions', 'register', 'registered', 'registrar', 'registration', 'registry', 'regress
ar', 'regularly', 'regulated', 'regulation', 'regulations', 'regulatory', 'rehab', 'rehabilitation', 'r
t', 'rejected', 'rel', 'relate', 'related', 'relates', 'relating', 'relation', 'relations', 'relationsh
onships', 'relative', 'relatively', 'relatives', 'relax', 'relaxation', 'relay', 'release', 'released',
'relevance', 'relevant', 'reliability', 'reliable', 'reliance', 'relief', 'religion', 'religions', 'rel
load', 'relocation', 'rely', 'relying', 'remain', 'remainder', 'remained', 'remaining', 'remains', 'rem
kable', 'remarks', 'remedies', 'remedy', 'remember', 'remembered', 'remind', 'reminder', 'remix', 'remo
ble', 'removal', 'remove', 'removed', 'removing', 'renaissance', 'render', 'rendered', 'rendering', 're
able', 'renewal', 'reno', 'rent', 'rental', 'rentals', 'rentcom', 'rep', 'repair', 'repairs', 'repeat',
'replace', 'replaced', 'replacement', 'replacing', 'replica', 'replication', 'replied', 'replies', 'rep
t', 'reported', 'reporter', 'reporters', 'reporting', 'reports', 'repository', 'represent', 'representa
esentations', 'representative', 'representatives', 'represented', 'representing', 'represents', 'reprin
s', 'reproduce', 'reproduced', 'reproduction', 'reproductive', 'republic', 'republican', 'republicans',
n', 'request', 'requested', 'requesting', 'requests', 'require', 'required', 'requirement', 'requiremen
es', 'requiring', 'res', 'rescue', 'research', 'researcher', 'researchers', 'reseller', 'reservation',
s', 'reserve', 'reserved', 'reserves', 'reservoir', 'reset', 'residence', 'resident', 'residential', 'r
'resist', 'resistance', 'resistant', 'resolution', 'resolutions', 'resolve', 'resolved', 'resort', 'res
urce', 'resources', 'respect', 'respected', 'respective', 'respectively', 'respiratory', 'respond', 're
espondent', 'respondents', 'responding', 'response', 'responses', 'responsibilities', 'responsibility',
e', 'rest', 'restaurant', 'restaurants', 'restoration', 'restore', 'restored', 'restrict', 'restricted'
on', 'restrictions', 'restructuring', 'result', 'resulted', 'resulting', 'results', 'resume', 'resumes'
'retailer', 'retailers', 'retain', 'retained', 'retention', 'retired', 'retirement', 'retreat', 'retrie
eve', 'retrieved', 'retro', 'return', 'returned', 'returning', 'returns', 'reunion', 'reuters', 'rev',
evealed', 'reveals', 'revelation', 'revenge', 'revenue', 'revenues', 'reverse', 'review', 'reviewed', '
```

```
'reviewing', 'reviews', 'revised', 'revision', 'revisions', 'revolution', 'revolutionary', 'reward', 'r
ynolds', 'rf', 'rfc', 'rg', 'rh', 'rhode', 'rhythm', 'ri', 'ribbon', 'rica', 'rice', 'rich', 'richard',
'richardson', 'richmond', 'rick', 'rico', 'rid', 'ride', 'rider', 'riders', 'rides', 'ridge', 'riding',
ights', 'rim', 'ring', 'rings', 'ringtone', 'ringtones', 'rio', 'rip', 'ripe', 'rise', 'rising', 'risk'
'river', 'rivers', 'riverside', 'rj', 'rl', 'rm', 'rn', 'rna', 'ro', 'road', 'roads', 'rob', 'robert',
'robertson', 'robin', 'robinson', 'robot', 'robots', 'robust', 'rochester', 'rock', 'rocket', 'rocks',
d', 'roger', 'rogers', 'roland', 'role', 'roles', 'roll', 'rolled', 'roller', 'rolling', 'rolls', 'rom'
'romance', 'romania', 'romantic', 'rome', 'ron', 'ronald', 'roof', 'room', 'roommate', 'roommates', 'ro
, 'roots', 'rope', 'rosa', 'rose', 'roses', 'ross', 'roster', 'rotary', 'rotation', 'rouge', 'rough', '
oulette', 'round', 'rounds', 'route', 'router', 'routers', 'routes', 'routine', 'routines', 'routing',
w', 'rows', 'roy', 'royal', 'royalty', 'rp', 'rpg', 'rpm', 'rr', 'rrp', 'rs', 'rss', 'rt', 'ru', 'rubbe
'rug', 'rugby', 'rugs', 'rule', 'ruled', 'rules', 'ruling', 'run', 'runner', 'running', 'runs', 'runtim
'rush', 'russell', 'russia', 'russian', 'ruth', 'rv', 'rw', 'rwanda', 'rx', 'ryan', 's', 'sa', 'sacrame
d', 'sacrifice', 'sad', 'saddam', 'safari', 'safe', 'safely', 'safer', 'safety', 'sage', 'sagem', 'said
'sailing', 'saint', 'saints', 'sake', 'salad', 'salaries', 'salary', 'sale', 'salem', 'sales', 'sally',
'salon', 'salt', 'salvador', 'salvation', 'sam', 'samba', 'same', 'samoa', 'sample', 'samples', 'sampli
g', 'samuel', 'san', 'sand', 'sandra', 'sandwich', 'sandy', 'sans', 'santa', 'sanyo', 'sao', 'sap', 'sa
ra', 'sarah', 'sas', 'saskatchewan', 'sat', 'satellite', 'satin', 'satisfaction', 'satisfactory', 'sati
isfy', 'saturday', 'saturn', 'sauce', 'saudi', 'savage', 'savannah', 'save', 'saved', 'saver', 'saves',
'savings', 'saw', 'say', 'saying', 'says', 'sb', 'sbjct', 'sc', 'scale', 'scales', 'scan', 'scanned', '
canners', 'scanning', 'scary', 'scenario', 'scenarios', 'scene', 'scenes', 'scenic', 'schedule', 'sched
dules', 'scheduling', 'schema', 'scheme', 'schemes', 'scholar', 'scholars', 'scholarship', 'scholarship
, 'schools', 'sci', 'science', 'sciences', 'scientific', 'scientist', 'scientists', 'scoop', 'scope', '
red', 'scores', 'scoring', 'scotia', 'scotland', 'scott', 'scottish', 'scout', 'scratch', 'screen', 'sc
creens', 'screensaver', 'screensavers', 'screenshot', 'screenshots', 'screw', 'script', 'scripting', 's
roll', 'scsi', 'scuba', 'sculpture', 'sd', 'se', 'sea', 'seafood', 'seal', 'sealed', 'sean', 'search',
'searched', 'searches', 'searching', 'seas', 'season', 'seasonal', 'seasons', 'seat', 'seating', 'seats
, 'sec', 'second', 'secondary', 'seconds', 'secret', 'secretariat', 'secretary', 'secrets', 'section',
'sector', 'sectors', 'secure', 'secured', 'securely', 'securities', 'security', 'see', 'seed', 'seeds',
'seek', 'seeker', 'seekers', 'seeking', 'seeks', 'seem', 'seemed', 'seems', 'seen', 'sees', 'sega', 'se
ments', 'select', 'selected', 'selecting', 'selection', 'selections', 'selective', 'self', 'sell', 'sel
rs', 'selling', 'sells', 'semester', 'semi', 'semiconductor', 'seminar', 'seminars', 'sen', 'senate', '
enators', 'send', 'sender', 'sending', 'sends', 'senegal', 'senior', 'seniors', 'sense', 'sensitive', '
, 'sensor', 'sensors', 'sent', 'sentence', 'sentences', 'seo', 'sep', 'separate', 'separated', 'separat
ation', 'sept', 'september', 'seq', 'sequence', 'sequences', 'ser', 'serbia', 'serial', 'series', 'seri
usly', 'serum', 'serve', 'served', 'server', 'servers', 'serves', 'service', 'services', 'serving', 'se
sions', 'set', 'sets', 'setting', 'settings', 'settle', 'settled', 'settlement', 'setup', 'seven', 'sev
ral', 'severe', 'sewing', 'sex', 'sexcam', 'sexo', 'sexual', 'sexuality', 'sexually', 'sexy', 'sf', 'sg
ade', 'shades', 'shadow', 'shadows', 'shaft', 'shake', 'shakespeare', 'shakira', 'shall', 'shame', 'sha
nnon', 'shape', 'shaped', 'shapes', 'share', 'shared', 'shareholders', 'shares', 'shareware', 'sharing'
'sharon', 'sharp', 'shaved', 'shaw', 'she', 'shed', 'sheep', 'sheer', 'sheet', 'sheets', 'sheffield', '
ll', 'shelter', 'shemale', 'shemales', 'shepherd', 'sheriff', 'sherman', 'shield', 'shift', 'shine', 's
ent', 'shipments', 'shipped', 'shipping', 'ships', 'shirt', 'shirts', 'shit', 'shock', 'shoe', 'shoes',
hooting', 'shop', 'shopper', 'shoppercom', 'shoppers', 'shopping', 'shoppingcom', 'shops', 'shopzilla',
hort', 'shortcuts', 'shorter', 'shortly', 'shorts', 'shot', 'shots', 'should', 'shoulder', 'show', 'sho
wed', 'shower', 'showers', 'showing', 'shown', 'shows', 'showtimes', 'shut', 'shuttle', 'si', 'sic', 's
, 'sides', 'sie', 'siemens', 'sierra', 'sig', 'sight', 'sigma', 'sign', 'signal', 'signals', 'signature
es', 'signed', 'significance', 'significant', 'significantly', 'signing', 'signs', 'signup', 'silence',
'silicon', 'silk', 'silly', 'silver', 'sim', 'similar', 'similarly', 'simon', 'simple', 'simplified', '
mpson', 'simpsons', 'sims', 'simulation', 'simulations', 'simultaneously', 'sin', 'since', 'sing', 'sin
nger', 'singh', 'singing', 'single', 'singles', 'sink', 'sip', 'sir', 'sister', 'sisters', 'sit', 'site
, 'sites', 'sitting', 'situated', 'situation', 'situations', 'six', 'sixth', 'size', 'sized', 'sizes',
ng', 'ski', 'skiing', 'skill', 'skilled', 'skills', 'skin', 'skins', 'skip', 'skirt', 'skirts', 'sku',
e', 'sl', 'slave', 'sleep', 'sleeping', 'sleeps', 'sleeve', 'slide', 'slides', 'slideshow', 'slight', '
'slim', 'slip', 'slope', 'slot', 'slots', 'slovak', 'slovakia', 'slovenia', 'slow', 'slowly', 'slut', '
, 'small', 'smaller', 'smart', 'smell', 'smile', 'smilies', 'smith', 'smithsonian', 'smoke', 'smoking',
'sms', 'smtp', 'sn', 'snake', 'snap', 'snapshot', 'snow', 'snowboard', 'so', 'soa', 'soap', 'soc', 'soc
l', 'societies', 'society', 'sociology', 'socket', 'socks', 'sodium', 'sofa', 'soft', 'softball', 'soft
l', 'sol', 'solar', 'solaris', 'sold', 'soldier', 'soldiers', 'sole', 'solely', 'solid', 'solo', 'solom
on', 'solutions', 'solve', 'solved', 'solving', 'soma', 'somalia', 'some', 'somebody', 'somehow', 'some
set', 'something', 'sometimes', 'somewhat', 'somewhere', 'son', 'song', 'songs', 'sonic', 'sons', 'sony
'soonest', 'sophisticated', 'sorry', 'sort', 'sorted', 'sorts', 'sought', 'soul', 'souls', 'sound', 'so
dtrack', 'soup', 'source', 'sources', 'south', 'southampton', 'southeast', 'southern', 'southwest', 'so
, 'sp', 'spa', 'space', 'spaces', 'spain', 'spam', 'span', 'spanish', 'spank', 'spanking', 'sparc', 'sp
, 'spatial', 'speak', 'speaker', 'speakers', 'speaking', 'speaks', 'spears', 'spec', 'special', 'specia
ialists', 'specialized', 'specializing', 'specially', 'specials', 'specialties', 'specialty', 'species'
, 'specifically', 'specification', 'specifications', 'specifics', 'specified', 'specifies', 'specify',
ectacular', 'spectrum', 'speech', 'speeches', 'speed', 'speeds', 'spell', 'spelling', 'spencer', 'spend
g', 'spent', 'sperm', 'sphere', 'spice', 'spider', 'spies', 'spin', 'spine', 'spirit', 'spirits', 'spir
rituality', 'split', 'spoke', 'spoken', 'spokesman', 'sponsor', 'sponsored', 'sponsors', 'sponsorship',
porting', 'sports', 'spot', 'spotlight', 'spots', 'spouse', 'spray', 'spread', 'spreading', 'spring', '
'springfield', 'springs', 'sprint', 'spy', 'spyware', 'sq', 'sql', 'squad', 'square', 'squirt', 'squirt
'src', 'sri', 'ss', 'ssl', 'st', 'stability', 'stable', 'stack', 'stadium', 'staff', 'staffing', 'stage
'stainless', 'stakeholders', 'stamp', 'stamps', 'stan', 'stand', 'standard', 'standards', 'standing', '
'stands', 'stanford', 'stanley', 'star', 'starring', 'stars', 'starsmerchant', 'start', 'started', 'sta
ting', 'starts', 'startup', 'stat', 'state', 'stated', 'statement', 'statements', 'states', 'statewide'
'stating', 'station', 'stationery', 'stations', 'statistical', 'statistics', 'stats', 'status', 'statut
s', 'statutory', 'stay', 'stayed', 'staying', 'stays', 'std', 'ste', 'steady', 'steal', 'steam', 'steel
g', 'stem', 'step', 'stephanie', 'stephen', 'steps', 'stereo', 'sterling', 'steve', 'steven', 'stevens'
'stick', 'sticker', 'stickers', 'sticks', 'sticky', 'still', 'stock', 'stockholm', 'stockings', 'stocks
'stomach', 'stone', 'stones', 'stood', 'stop', 'stopped', 'stopping', 'stops', 'storage', 'store', 'sto
s', 'stories', 'storm', 'story', 'str', 'straight', 'strain', 'strand', 'strange', 'stranger', 'strap',
, 'strategies', 'strategy', 'stream', 'streaming', 'streams', 'street', 'streets', 'strength', 'strengt
gthening', 'strengths', 'stress', 'stretch', 'strict', 'strictly', 'strike', 'strikes', 'striking', 'st
ngs', 'strip', 'stripes', 'strips', 'stroke', 'strong', 'stronger', 'strongly', 'struck', 'struct', 'st
'structure', 'structured', 'structures', 'struggle', 'stuart', 'stuck', 'stud', 'student', 'students',
'studies', 'studio', 'studios', 'study', 'studying', 'stuff', 'stuffed', 'stunning', 'stupid', 'style',
'stylish', 'stylus', 'su', 'sub', 'subaru', 'subcommittee', 'subdivision', 'subject', 'subjects', 'subl
medirectory', 'submission', 'submissions', 'submit', 'submitted', 'submitting', 'subscribe', 'subscribe
```

```
medirectory', 'submission', 'submissions', 'submit', 'submitted', 'submitting', 'subscribe', 'subscribe
bers', 'subscription', 'subscriptions', 'subsection', 'subsequent', 'subsequently', 'subsidiaries', 'su
'substance', 'substances', 'substantial', 'substantially', 'substitute', 'subtle', 'suburban', 'succeed
, 'successful', 'successfully', 'such', 'suck', 'sucking', 'sucks', 'sudan', 'sudden', 'suddenly', 'sue
'suffered', 'suffering', 'sufficient', 'sufficiently', 'sugar', 'suggest', 'suggested', 'suggesting', '
'suggestions', 'suggests', 'suicide', 'suit', 'suitable', 'suite', 'suited', 'suites', 'suits', 'sulliv
'summaries', 'summary', 'summer', 'summit', 'sun', 'sunday', 'sunglasses', 'sunny', 'sunrise', 'sunset'
, 'super', 'superb', 'superintendent', 'superior', 'supervision', 'supervisor', 'supervisors', 'supplem
emental', 'supplements', 'supplied', 'supplier', 'suppliers', 'supplies', 'supply', 'support', 'support
ters', 'supporting', 'supports', 'suppose', 'supposed', 'supreme', 'sur', 'sure', 'surely', 'surf', 'su
faces', 'surfing', 'surge', 'surgeon', 'surgeons', 'surgery', 'surgical', 'surname', 'surplus', 'surpri
sed', 'surprising', 'surrey', 'surround', 'surrounded', 'surrounding', 'surveillance', 'survey', 'surve
al', 'survive', 'survivor', 'survivors', 'susan', 'suse', 'suspect', 'suspected', 'suspended', 'suspens
x', 'sustainability', 'sustainable', 'sustained', 'suzuki', 'sv', 'sw', 'swap', 'sweden', 'swedish', 's
t', 'swim', 'swimming', 'swing', 'swingers', 'swiss', 'switch', 'switched', 'switches', 'switching', 's
'sword', 'sydney', 'symantec', 'symbol', 'symbols', 'sympathy', 'symphony', 'symposium', 'symptoms', 's
cate', 'syndication', 'syndrome', 'synopsis', 'syntax', 'synthesis', 'synthetic', 'syracuse', 'syria', 
em', 'systematic', 'systems', 't', 'ta', 'tab', 'table', 'tables', 'tablet', 'tablets', 'tabs', 'tackle
, 'tag', 'tagged', 'tags', 'tahoe', 'tail', 'taiwan', 'take', 'taken', 'takes', 'taking', 'tale', 'tale
ed', 'tales', 'talk', 'talked', 'talking', 'talks', 'tall', 'tamil', 'tampa', 'tan', 'tank', 'tanks', '
'tap', 'tape', 'tapes', 'tar', 'target', 'targeted', 'targets', 'tariff', 'task', 'tasks', 'taste', 'ta
ht', 'tax', 'taxation', 'taxes', 'taxi', 'taylor', 'tb', 'tba', 'tc', 'tcp', 'td', 'te', 'tea', 'teach'
'teachers', 'teaches', 'teaching', 'team', 'teams', 'tear', 'tears', 'tech', 'technical', 'technician',
, 'techniques', 'techno', 'technological', 'technologies', 'technology', 'techrepublic', 'ted', 'teddy'
en', 'teenage', 'teens', 'teeth', 'tel', 'telecharger', 'telecom', 'telecommunications', 'telephone', '
'telescope', 'television', 'televisions', 'tell', 'telling', 'tells', 'temp', 'temperature', 'temperatu
ate', 'templates', 'temple', 'temporal', 'temporarily', 'temporary', 'ten', 'tenant', 'tend', 'tender',
, 'tennis', 'tension', 'tent', 'term', 'terminal', 'terminals', 'termination', 'terminology', 'terms', 
'terrain', 'terrible', 'territories', 'territory', 'terror', 'terrorism', 'terrorist', 'terrorists', 't
t', 'testament', 'tested', 'testimonials', 'testimony', 'testing', 'tests', 'tex', 'texas', 'text', 'te
xtbooks', 'textile', 'textiles', 'texts', 'texture', 'tf', 'tft', 'tgp', 'th', 'thai', 'thailand', 'tha
'thanks', 'thanksgiving', 'that', 'thats', 'the', 'theater', 'theaters', 'theatre', 'thee', 'theft', 't
ir', 'them', 'theme', 'themes', 'themselves', 'then', 'theology', 'theorem', 'theoretical', 'theories',
'therapeutic', 'therapist', 'therapy', 'there', 'thereafter', 'thereby', 'therefore', 'thereof', 'therm
rus', 'these', 'thesis', 'they', 'thick', 'thickness', 'thin', 'thing', 'things', 'think', 'thinking', 
'thinks', 'third', 'thirty', 'this', 'thomas', 'thompson', 'thomson', 'thong', 'thongs', 'thorough', 't
'those', 'thou', 'though', 'thought', 'thoughts', 'thousand', 'thousands', 'thread', 'threaded', 'threa
t', 'threatened', 'threatening', 'threats', 'three', 'threesome', 'threshold', 'thriller', 'throat', 't
roughout', 'throw', 'throwing', 'thrown', 'throws', 'thru', 'thu', 'thumb', 'thumbnail', 'thumbnails', 
humbzilla', 'thunder', 'thursday', 'thus', 'thy', 'ti', 'ticket', 'tickets', 'tide', 'tie', 'tied', 'ti
'tiffany', 'tiger', 'tigers', 'tight', 'til', 'tile', 'tiles', 'till', 'tim', 'timber', 'time', 'timeli
y', 'timer', 'times', 'timing', 'timothy', 'tin', 'tiny', 'tion', 'tions', 'tip', 'tips', 'tire', 'tire
'tissue', 'tit', 'titanium', 'titans', 'title', 'titled', 'titles', 'tits', 'titten', 'tm', 'tmp', 'tn'
acco', 'tobago', 'today', 'todd', 'toddler', 'toe', 'together', 'toilet', 'token', 'tokyo', 'told', 'to
oll', 'tom', 'tomato', 'tomatoes', 'tommy', 'tomorrow', 'ton', 'tone', 'toner', 'tones', 'tongue', 'ton
s', 'tony', 'too', 'took', 'tool', 'toolbar', 'toolbox', 'toolkit', 'tools', 'tooth', 'top', 'topic', '
pless', 'tops', 'toronto', 'torture', 'toshiba', 'total', 'totally', 'totals', 'touch', 'touched', 'tou
'touring', 'tourism', 'tourist', 'tournament', 'tournaments', 'tours', 'toward', 'towards', 'tower', 't
n', 'towns', 'township', 'toxic', 'toy', 'toyota', 'toys', 'tp', 'tr', 'trace', 'track', 'trackback', '
'tracked', 'tracker', 'tracking', 'tracks', 'tract', 'tractor', 'tracy', 'trade', 'trademark', 'tradema
r', 'trades', 'trading', 'tradition', 'traditional', 'traditions', 'traffic', 'tragedy', 'trail', 'trai
ers', 'trails', 'train', 'trained', 'trainer', 'trainers', 'training', 'trains', 'tramadol', 'trance', 
rans', 'transaction', 'transactions', 'transcript', 'transcription', 'transcripts', 'transexual', 'tran
'transfer', 'transferred', 'transfers', 'transform', 'transformation', 'transit', 'transition', 'transl
lated', 'translation', 'translations', 'translator', 'transmission', 'transmit', 'transmitted', 'transp
ansparent', 'transport', 'transportation', 'transsexual', 'trap', 'trash', 'trauma', 'travel', 'travele
rs', 'traveling', 'traveller', 'travelling', 'travels', 'travesti', 'travis', 'tray', 'treasure', 'trea
asures', 'treasury', 'treat', 'treated', 'treating', 'treatment', 'treatments', 'treaty', 'tree', 'tree
'trembl', 'tremendous', 'trend', 'trends', 'treo', 'tri', 'trial', 'trials', 'triangle', 'tribal', 'tri
s', 'tribunal', 'tribune', 'tribute', 'trick', 'tricks', 'tried', 'tries', 'trigger', 'trim', 'trinidad
, 'trio', 'trip', 'tripadvisor', 'triple', 'trips', 'triumph', 'trivia', 'troops', 'tropical', 'trouble
hooting', 'trout', 'troy', 'truck', 'trucks', 'true', 'truly', 'trunk', 'trust', 'trusted', 'trustee', 
'trusts', 'truth', 'try', 'trying', 'ts', 'tsunami', 'tt', 'tu', 'tub', 'tube', 'tubes', 'tucson', 'tue
, 'tuition', 'tulsa', 'tumor', 'tune', 'tuner', 'tunes', 'tuning', 'tunisia', 'tunnel', 'turbo', 'turke
h', 'turn', 'turned', 'turner', 'turning', 'turns', 'turtle', 'tutorial', 'tutorials', 'tv', 'tvcom', '
e', 'twenty', 'twice', 'twiki', 'twin', 'twinks', 'twins', 'twist', 'twisted', 'two', 'tx', 'ty', 'tyle
'types', 'typical', 'typically', 'typing', 'u', 'uc', 'uganda', 'ugly', 'uh', 'ui', 'uk', 'ukraine', 'u
e', 'ultimately', 'ultra', 'ultram', 'um', 'un', 'una', 'unable', 'unauthorized', 'unavailable', 'uncer
cle', 'und', 'undefined', 'under', 'undergraduate', 'underground', 'underlying', 'understand', 'underst
derstood', 'undertake', 'undertaken', 'underwear', 'undo', 'une', 'unemployment', 'unexpected', 'unfort
ni', 'unified', 'uniform', 'union', 'unions', 'uniprotkb', 'unique', 'unit', 'united', 'units', 'unity'
niversal', 'universe', 'universities', 'university', 'unix', 'unknown', 'unless', 'unlike', 'unlikely',
, 'unlock', 'unnecessary', 'unsigned', 'unsubscribe', 'until', 'untitled', 'unto', 'unusual', 'unwrap',
, 'upcoming', 'update', 'updated', 'updates', 'updating', 'upgrade', 'upgrades', 'upgrading', 'upload',
'upon', 'upper', 'ups', 'upset', 'upskirt', 'upskirts', 'ur', 'urban', 'urge', 'urgent', 'uri', 'url', 
guay', 'urw', 'us', 'usa', 'usage', 'usb', 'usc', 'usd', 'usda', 'use', 'used', 'useful', 'user', 'user
s', 'uses', 'usgs', 'using', 'usps', 'usr', 'usual', 'usually', 'ut', 'utah', 'utc', 'utilities', 'util
zation', 'utilize', 'utils', 'uv', 'uw', 'uzbekistan', 'v', 'va', 'vacancies', 'vacation', 'vacations',
'vacuum', 'vagina', 'val', 'valentine', 'valid', 'validation', 'validity', 'valium', 'valley', 'valuabl
on', 'value', 'valued', 'values', 'valve', 'valves', 'vampire', 'van', 'vancouver', 'vanilla', 'var', '
'variables', 'variance', 'variation', 'variations', 'varied', 'varies', 'variety', 'various', 'vary', '
ast', 'vat', 'vatican', 'vault', 'vb', 'vbulletin', 'vc', 'vcr', 've', 'vector', 'vegas', 'vegetable', 
, 'vegetarian', 'vegetation', 'vehicle', 'vehicles', 'velocity', 'velvet', 'vendor', 'vendors', 'venezu
e', 'venture', 'ventures', 'venue', 'venues', 'ver', 'verbal', 'verde', 'verification', 'verified', 've
zon', 'vermont', 'vernon', 'verse', 'version', 'versions', 'versus', 'vertex', 'vertical', 'very', 'ver
'vessel', 'vessels', 'veteran', 'veterans', 'veterinary', 'vg', 'vhs', 'vi', 'via', 'viagra', 'vibrator
s', 'vic', 'vice', 'victim', 'victims', 'victor', 'victoria', 'victorian', 'victory', 'vid', 'video', '
ds', 'vienna', 'vietnam', 'vietnamese', 'view', 'viewed', 'viewer', 'viewers', 'viewing', 'viewpicture'
'vii', 'viii', 'viking', 'villa', 'village', 'villages', 'villas', 'vincent', 'vintage', 'vinyl', 'viol
lations', 'violence', 'violent', 'violin', 'vip', 'viral', 'virgin', 'virginia', 'virtual', 'virtually'
```

```
lations', 'violence', 'violent', 'violin', 'vip', 'viral', 'virgin', 'virginia', 'virtual', 'virtually
'virus', 'viruses', 'visa', 'visibility', 'visible', 'vision', 'visit', 'visited', 'visiting', 'visitor
s', 'visits', 'vista', 'visual', 'vital', 'vitamin', 'vitamins', 'vocabulary', 'vocal', 'vocals', 'voca
ice', 'voices', 'void', 'voip', 'vol', 'volkswagen', 'volleyball', 'volt', 'voltage', 'volume', 'volume
ry', 'volunteer', 'volunteers', 'volvo', 'von', 'vote', 'voted', 'voters', 'votes', 'voting', 'voyeur',
, 'voyuer', 'vp', 'vpn', 'vs', 'vsnet', 'vt', 'vulnerability', 'vulnerable', 'w', 'wa', 'wage', 'wages'
'wagon', 'wait', 'waiting', 'waiver', 'wake', 'wal', 'wales', 'walk', 'walked', 'walker', 'walking', 'w
l', 'wallace', 'wallet', 'wallpaper', 'wallpapers', 'walls', 'walnut', 'walt', 'walter', 'wan', 'wang',
ant', 'wanted', 'wanting', 'wants', 'war', 'warcraft', 'ward', 'ware', 'warehouse', 'warm', 'warming',
arner', 'warning', 'warnings', 'warrant', 'warranties', 'warranty', 'warren', 'warrior', 'warriors', 'w
'wash', 'washer', 'washing', 'washington', 'waste', 'watch', 'watched', 'watches', 'watching', 'water',
f', 'waters', 'watershed', 'watson', 'watt', 'watts', 'wav', 'wave', 'waves', 'wax', 'way', 'wayne', 'w
'wc', 'we', 'weak', 'wealth', 'weapon', 'weapons', 'wear', 'wearing', 'weather', 'web', 'webcam', 'webc
st', 'weblog', 'weblogs', 'webmaster', 'webmasters', 'webpage', 'webshots', 'website', 'websites', 'web
, 'wedding', 'weddings', 'wednesday', 'weed', 'week', 'weekend', 'weekends', 'weekly', 'weeks', 'weight
d', 'weights', 'weird', 'welcome', 'welding', 'welfare', 'well', 'wellington', 'wellness', 'wells', 'we
y', 'went', 'were', 'wesley', 'west', 'western', 'westminster', 'wet', 'whale', 'what', 'whatever', 'wh
t', 'wheel', 'wheels', 'when', 'whenever', 'where', 'whereas', 'wherever', 'whether', 'which', 'while',
'white', 'who', 'whole', 'wholesale', 'whom', 'whore', 'whose', 'why', 'wi', 'wichita', 'wicked', 'wide
'wider', 'widescreen', 'widespread', 'width', 'wife', 'wifi', 'wiki', 'wikipedia', 'wild', 'wilderness'
, 'wiley', 'will', 'william', 'williams', 'willing', 'willow', 'wilson', 'win', 'wind', 'window', 'wind
s', 'windsor', 'wine', 'wines', 'wing', 'wings', 'winner', 'winners', 'winning', 'wins', 'winston', 'wi
e', 'wired', 'wireless', 'wires', 'wiring', 'wisconsin', 'wisdom', 'wise', 'wish', 'wishes', 'wishlist'
tch', 'with', 'withdrawal', 'within', 'without', 'witness', 'witnesses', 'wives', 'wizard', 'wm', 'wma'
f', 'woman', 'women', 'womens', 'won', 'wonder', 'wonderful', 'wondering', 'wood', 'wooden', 'woods', '
ester', 'word', 'wordpress', 'words', 'work', 'worked', 'worker', 'workers', 'workflow', 'workforce', '
orkout', 'workplace', 'works', 'workshop', 'workshops', 'workstation', 'world', 'worldcat', 'worlds', '
'worldwide', 'worm', 'worn', 'worried', 'worry', 'worse', 'worship', 'worst', 'worth', 'worthy', 'would
'wow', 'wp', 'wr', 'wrap', 'wrapped', 'wrapping', 'wrestling', 'wright', 'wrist', 'write', 'writer', 'w
ites', 'writing', 'writings', 'written', 'wrong', 'wrote', 'ws', 'wt', 'wto', 'wu', 'wv', 'ww', 'www',
'wyoming', 'x', 'xanax', 'xbox', 'xerox', 'xhtml', 'xi', 'xl', 'xml', 'xnxx', 'xp', 'xx', 'xxx', 'y', '
, 'yahoo', 'yale', 'yamaha', 'yang', 'yard', 'yards', 'yarn', 'ye', 'yea', 'yeah', 'year', 'yearly', 'y
t', 'yellow', 'yemen', 'yen', 'yes', 'yesterday', 'yet', 'yield', 'yields', 'yn', 'yo', 'yoga', 'york',
, 'you', 'young', 'younger', 'your', 'yours', 'yourself', 'youth', 'yr', 'yrs', 'yu', 'yugoslavia', 'yu
'za', 'zambia', 'zdnet', 'zealand', 'zen', 'zero', 'zimbabwe', 'zinc', 'zip', 'zoloft', 'zone', 'zones'
'zoo', 'zoom', 'zoophilia', 'zope', 'zshops', 'zu', 'zum', 'zus', "'"]
```

Exercise 0 (10 points). Your task is to write a function that takes any string as an input, and returns a list of any anagrams---that is, words made same letters---for that string that appear on the common words list.

If the string happens to be a word from the common words list, it should appear in the output. So, for instance, if the input were the string `"weir` should be `['weird', 'wider', 'wired']`. If there are no anagrams for the given input, your function should return an empty list.

> **Bonus challenge.** Can you augment your code to find the longest anagram pair in the common words list? What about the longest ana triple, i.e., three words with exactly the same letters?

In [ ]:
```
def anagram_check(word):
    assert type(word) is str
    ### BEGIN SOLUTION
    from collections import defaultdict
    word_dic = defaultdict(list)
    for aword in commonwordslist:
        word_dic[''.join(sorted(aword))].append(aword)
    sorted_word = ''.join(sorted(word))
    return word_dic[sorted_word]
    ### END SOLUTION
```

In [ ]:
```
# Test cell: test_dummy

assert set(anagram_check("weird")) == {'weird', 'wider', 'wired'}
assert set(anagram_check("aegllry")) == {'allergy', 'gallery', 'largely'}
assert anagram_check("this string is way too long to possibly have a match") == []

print("\n(Passed!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 7

Letter frequencies. This problem has three (3) exercises worth a total of ten (10) points.

Letter frequency in text has been studied in cryptoanalysis, in particular frequency analysis. Linguists use letter frequency analysis as a rudimen for language identification, where it's particularly effective as an indicator of whether an unknown writing system is alphabetic, syllablic, or ideogı

Primarily, three different ways exist for letter frequency analysis. Each way generally results in very different charts for common letters. Based or text, the first method is to count letter frequency in root words of a dictionary. The second way is to include all word variants when counting, suck going and goes and not just the root word go. Such a system results in letters like "s" appearing much more frequently. The last variant is to cou based on their frequency in the actual text that is being studied.

For more details, refer to the link: https://en.wikipedia.org/wiki/Letter_frequency (https://en.wikipedia.org/wiki/Letter_frequency)

In this problem, we will focus on the 3rd methodology.

Exercise 0 (2 points). First, given a string input, define a function `preprocess` that returns a string with non-alphabetic characters removed and alphabets converted into a lower case.

For example, 'We are coding letter Frequency! Yay!" would be transformed into "wearecodingletterfrequencyyay"

In [ ]:

```
def preprocess(S):
    ### BEGIN SOLUTION
    s = ''.join([c.lower() for c in S if c.isalpha()])
    return s
    ### END SOLUTION
```

In [ ]:

```
# Test cell: valid_string
import random, string

N_str = 100 #Length of random string

def generate_str(n):
    random_str = ''.join(random.choice(string.ascii_lowercase + string.ascii_uppercase + string.digits
ctuation) for _ in range(n))
    return random_str

def check_preprocess_str(n):
    random_str = generate_str(n)
    print("Input String: ",random_str)
    assert preprocess('random_str').islower() == True
    assert preprocess(random_str).isalpha() == True
    print("|----Your function seems to work correct for the string----|"+"\n")

check_preprocess_str(N_str)
check_preprocess_str(N_str)
check_preprocess_str(N_str)

print("\n(Passed)!")
```

Exercise 1 (4 points). With the necessary pre-processing complete, the next step is to write a function `count_letters(S)` to count the number of each letter in the alphabet.

You can assume that only letters will be present in the input string. It should output a dictionary and if any alphabet (a-z) is missing in the input s still be a part of the output dictionary and its corresponding value should be equal to zero.

In [ ]:

```
def count_letters(S):
    ### BEGIN SOLUTION
    alphabets = string.ascii_lowercase
    count_dict = {c:S.count(c) for c in alphabets}
    count_dict = {c:S.count(c) for c in alphabets}
    return count_dict
    ### END SOLUTION
```

In [ ]:

```
# Test cell: count_letters
import collections

N_processed_str = 100

def generate_processed_str(n):
    random_processed_str = ''.join(random.choice(string.ascii_lowercase) for _ in range(n))
    return random_processed_str

def check_count_letters(S):
    print("Input String: ",S)
    random_char = chr(random.randint(97,122))
    print("Character frequency evaluated for: ", random_char)
    if(random_char in S):
        assert count_letters(S)[random_char] == collections.Counter(S)[random_char]
        print("|----Your function seems to return correct freq for the char----|"+"\n")
    else:
        assert count_letters(S)[random_char] == 0
        print("|----Your function seems to return correct freq for the char----|"+"\n")

check_count_letters(generate_processed_str(N_processed_str))
check_count_letters(generate_processed_str(N_processed_str))
check_count_letters(generate_processed_str(N_processed_str))
print("\n(Passed)!")
```

Exercise 2 (4 points). The next step is to sort the distribution of a dictionary containing all the letters in the alphabet as keys and number of occu as associated value.

Sorting should be first done in decreasing order by occurrence count and for two elements with same count, the order should be alphabetic. The `find_top_letter(d)` should return the 1st character in the order.

In [ ]:

```
def find_top_letter(d):
    ### BEGIN SOLUTION
    t = [(l, o) for l,o in d.items()]
    t.sort(key = lambda x: (x[1]*-1, x[0]))
    return t[:1][0][0]
    ### END SOLUTION
```

In [ ]:

```
# Test cell: highest_freq_letter

def create_random_dict():
    max_char_value = random.randint(5, 20)
    random_dict = {c:random.randint(0,max_char_value-1) for c in string.ascii_lowercase}
    random_letter1, random_letter2 = random.sample(string.ascii_lowercase, 2)
    random_dict[random_letter1], random_dict[random_letter2] = max_char_value, max_char_value
    if(random_letter1 < random_letter2):
        return random_letter1, random_dict
    else:
        return random_letter2, random_dict

def check_top_letter():
    top_letter, random_dict = create_random_dict()
    user_letter = find_top_letter(random_dict)
    assert user_letter == top_letter
    print("Input Dictionary: ", random_dict)
    print("Your function correctly returned most frequent letter: {} \n".format(user_letter))

check_top_letter()
check_top_letter()
check_top_letter()
print("\n(Passed)!")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 8

This problem has three (3) exercises worth a total of ten (10) points.

Invert a dictionary. Suppose a group of student TAs have been asked to monitor the Piazza forums for your CSE6040x course. The assignmer follows.

In [ ]:
```
assignments = {"Monday" : ["Jeh", "Ben", "Chinmay", "Rachel"],
               "Tuesday" : ["Rachel", "Samuel", "Evan", "Raghav"],
               "Wednesday" : ["Evan", "Samuel", "Michael", "Chinmay"],
               "Thursday" : ["Jeh", "Michael", "Ben", "Evan"],
               "Friday" : ["Rachel", "Shishir", "Samuel", "Ben"],
               "Saturday" : ["Yandi", "Michael", "Chinmay", "Samuel"],
               "Sunday" : ["Evan", "Chinmay", "Jeh", "Rachel"]
               }
```

Exercise 0 (5 points). Write a function, `invert_dict(d)`, that "inverts" a given dictionary `d`. The output should be another dictionary such that tl TA names and their corresponding values should be a list which stores the days assigned to them.

For example, one of the keys of the returned dictionary should be `"Jeh"` with its corresponding list as `["Monday", "Thursday", "Sunday"]`.

In [ ]:
```
from collections import defaultdict
def invert_dict(d):
    ### BEGIN SOLUTION
    l = []
    for k, v in d.items():
        l.append(v)
    flatten_l = [i for e in l for i in e]
    flatten_l = set(flatten_l)
    d_inv = defaultdict(list)
    for e in flatten_l:
        for k, v in d.items():
            if e in v:
                d_inv[e].append(k)
    return dict(d_inv)
    ### END SOLUTION
```

In [ ]:
```
# Test cell: `test_inverse`

import random
TA_list = ["Rachel", "Yandi", "Ben", "Jeh", "Evan", "Chinmay", "Shishir", "Raghav", "Samuel", "Michael"

# Your solution
inv_dict = invert_dict(assignments)

random_TA = random.sample(TA_list, 5)
for TA in random_TA:
    assigned_days = inv_dict[TA]
    for days in assigned_days:
        assert TA in assignments[days], "Incorrect inversion for TA {}".format(TA)

print("\n(Passed!)")
```

Exercise 2 (2 points): *Tracing the route of a pilot*.

Suppose a pilot flies from one city to the next. He makes 7 such trips, which are stored in the list shown in the next cell. The first entry of each tu the origin, and the second denotes the destination. Also assume that the next flight a pilot makes must originate from the destination of her prev task in this exercise is to write a function that finds the route followed by the pilot, given her first port of origin.

First, let `segments` be an unordered list of segments that the pilot flew.

In [ ]:
```
segments = [("JFK", "DEN"), ("LAX", "ORD"), ("DEN", "SFO"), ("LAS", "LAX"), ("ORD", "ATL"), ("ATL", "JF
"LAS")]
```

Next, write a function `find_dest(segs, origin)` that returns the next destination of the pilot given one port of origin and an unordered list of fl Example, if `segs == segments` as defined above, then for `'LAX'`, your function should return `'ORD'` because there is the tuple, `("LAX", "ORD'`

You may assume that `origin` appears only once in the list of segments, `segs`.

In [ ]:
```
def find_dest(segs, origin):
    assert type(origin) is str
    assert type(segs) is list
    ### BEGIN SOLUTION
    for i in segs:
        if i[0] == origin:
            dest = i[1]
            return dest
    ### END SOLUTION
```

In [ ]:
```
# Test flight destination
den = find_dest(segments, "DEN")
assert find_dest(segments, "DEN") == "SFO", "Wrong destination for DEN"
assert find_dest(segments, "LAX") == "ORD", "Wrong destination for LAX"
assert find_dest(segments, "ATL") == "JFK", "Wrong destination for ATL"
assert find_dest(segments, "JFK") == "DEN", "Wrong destination for JFK"
print("\n(Passed.)")
```

Exercise 3 (3 points). Now write a function `get_route(segs, origin)` that traces the complete itinerary of the pilot, given her first port of origi the destination for a flight is the same as the origin for the next flight. The itinerary is completed when he arrives back at the starting port of origi example, if the starting port is `"JFK"`, your function should return the list: `["JFK", "DEN", "SFO", "LAS", "LAX", "ORD", "ATL"]`.

In [ ]:
```
def get_route(path, origin):
    ### BEGIN SOLUTION
    route = [origin]
    dest = find_dest(path, origin)
    while dest != origin:
        route.append(dest)
        dest = find_dest(path, dest)
    return route
    ### END SOLUTION
```

In [ ]:
```
# Test pilot route
ports = ["JFK", "DEN", "SFO", "LAS", "LAX", "ORD", "ATL"]
starting_port = random.sample(ports, 5)
for p in starting_port:
    itinerary = get_route(segments, p)
    assert itinerary[0] == p, "incorrect port of origin for the itinerary"
    for i, port in enumerate(itinerary[:-1]):
        dest = find_dest(segments, port)
        assert dest == itinerary[i+1], "incorrect itinerary"

print("\n(Passed.)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 9

This problem consists of a single exercise worth ten (10) points.

Exercise 0 (10 points): Write a funtion, `digitize(n)`, that, given a postive number `n`, returns the digits of `n` as the integer elements of a list in re

For example:

```
assert digitize(1345328) == [8, 2, 3, 5, 4, 3, 1]
```

In [ ]:
```
def digitize(n):
    assert n > 0, '`n` is not a positive number'
    assert type(n) is int, '`n` must be an `int`'
    ### BEGIN SOLUTION
    digits = [int(d) for d in str(n)]
    return digits[::-1]
    ### END SOLUTION
```

In [ ]:
```
# Test_cell: test_digitize

assert digitize(1345328) == [8, 2, 3, 5, 4, 3, 1], 'Wrong output. Make sure your function returns the d
erse order'
print("\n(Passed!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

Problem 10: Tic-Tac-Toe!

The objective of this exercise is to check whether you can declare a winner based on the current state of a Tic-Tac-Toe (TTT) game (https://en.wikipedia.org/wiki/Tic-tac-toe (https://en.wikipedia.org/wiki/Tic-tac-toe)).

A state in the game of TTT can be represented as a list of lists as follows

1. Crosses are represented by -1
2. Circles are represented by +1
3. Unused cells are represented by 0

For example, the game below

	col0	col1	col2
row0	0	0	-1
row1	1	-1	1
row2	-1	1	0

can be represented as a list of lists

```
[[0,0,-1],
 [1,-1,1],
 [-1,1,0]]
```

Exercise 0 Write a function called `check_ttt(game)` that takes a list of 3 lists representing each row of a game's state and determines whether a winner. In particular, it should return one of the following strings:

1. `"Circles (+1) Win"`
2. `"Crosses (-1) Win"`
3. `"No Result"`

For example,

```
assert check_ttt([[0,0,-1],[1,-1,1],[-1,1,0]]) == "Crosses (-1) Win"
```

since there are three crosses (-1 values) along one of the diagonals of this board state.

You may assume that the input is a valid game state, i.e., a list representing a 3×3 grid with all cells having values in $\{-1, 0, 1\}$ and at most winner.

In [ ]:

```
def check_ttt(game):
    # Check that the input is valid
    assert type(game) == list
    assert all([type(row) == list for row in game])
    assert all([[elem in [0,1,-1] for elem in row] for row in game])

    ### BEGIN SOLUTION
    # Check for horizontal and vertical sums
    for num in range(3):
        hor_sum = sum(game[num])
        vert_sum = sum([vert[num] for vert in game])
        if max(hor_sum,vert_sum)==3: return "Circles (+1) Win"
        elif min(hor_sum,vert_sum)==-3: return "Crosses (-1) Win"

    # Check for diagonals
    diag_for = sum([diag[i] for i,diag in enumerate(game)])
    diag_back = sum([diag[2-i] for i,diag in enumerate(game)])
    if max(diag_for,diag_back)==3: return "Circles (+1) Win"
    elif min(diag_for,diag_back)==-3: return "Crosses (-1) Win"

    return "No Result"
    ### END SOLUTION
```

In [ ]:

```
# Test cell: test_cell_part0
import random
import numpy as np

# test 1
test_game1 = [[0,0,0] for elem in range(3)]
assert check_ttt(test_game1) == "No Result"

# test 2
test_game2 = [[1,1,1],[-1,-1,0],[0,0,0]]
assert check_ttt(test_game2) == "Circles (+1) Win"
```

```
# test 3
test_game3 = [[0,0,-1],[1,-1,1],[-1,1,0]]
assert check_ttt(test_game3) == "Crosses (-1) Win"

# test 4
test_game4 = [[0,0,-1],[1,-1,-1],[1,1,-1]]
assert check_ttt(test_game4) == "Crosses (-1) Win"

print("\n(Passed!)")
```

In [ ]:

```
# Test cell: test_cell_part1
import random
import numpy as np


def test_check_ttt():
    test_game_random = [[0,0,0] for elem in range(3)]
    boxes_filled = random.randint(0,8)
    num_circles = int(boxes_filled/2)

    circle_ind = random.sample(range(9),num_circles)
    cross_ind = random.sample([elem for elem in range(9) if elem not in circle_ind],(boxes_filled-num_c

    for elem in circle_ind: test_game_random[int(elem/3)][elem%3] = 1
    for elem in cross_ind: test_game_random[int(elem/3)][elem%3] = -1
    print(np.array(test_game_random))
    print("\nYour Result: {}".format(check_ttt(test_game_random)))

    your_answer = check_ttt(test_game_random)
    if 3 in np.sum(test_game_random,axis=0) & -3 in np.sum(test_game_random,axis=0): pass
    elif 3 in np.sum(test_game_random,axis=1) & -3 in np.sum(test_game_random,axis=1): pass
    elif 3 in np.sum(test_game_random,axis=0) & -3 in np.sum(test_game_random,axis=1): pass
    elif 3 in np.sum(test_game_random,axis=0) & -3 in np.sum(test_game_random,axis=1): pass
    elif 3 in np.sum(test_game_random,axis=0): assert your_answer == "Circles (+1) Win"
    elif -3 in np.sum(test_game_random,axis=0): assert your_answer == "Crosses (-1) Win"
    elif 3 in np.sum(test_game_random,axis=1): assert your_answer == "Circles (+1) Win"
    elif -3 in np.sum(test_game_random,axis=1): assert your_answer == "Crosses (-1) Win"
    elif np.trace(test_game_random) == 3: assert your_answer == "Circles (+1) Win"
    elif np.trace(test_game_random) == -3: assert your_answer == "Crosses (-1) Win"
    elif np.flipud(test_game_random).trace() == 3: assert your_answer == "Circles (+1) Win"
    elif np.flipud(test_game_random).trace() == -3: assert your_answer == "Crosses (-1) Win"
    else: assert check_ttt(test_game_random) == "No Result"

print("testing function for random game states..")
for i in range(20):
    print("---------- Game State {} ----------".format(i))
    test_check_ttt()

print("\n(Passed!)")
```

Fin! You've reached the end of this problem. Don't forget to restart the kernel and run the entire notebook from top-to-bottom to make sure you c correctly. If that is working, try submitting this problem. (Recall that you *must* submit and pass the autograder to get credit for your work!)

‹ Previous

Next Up: Topic 2: Pairwise Association Mining ›
36 min