**Function Practice Exercises**

Problems are arranged in increasing difficulty:

* Warmup - these can be solved using basic comparisons and methods
* Level 1 - these may involve if/then conditional statements and simple methods
* Level 2 - these may require iterating over sequences, usually with some kind of loop
* Challenging - these will take some creativity to solve

**WARMUP SECTION:**

**Problem 1:**

LESSER OF TWO EVENS: Write a function that returns the lesser of two given numbers *if* both numbers are even, but returns the greater if one or both numbers are odd lesser\_of\_two\_evens(2,4) --> 2 lesser\_of\_two\_evens(2,5) --> 5

**Problem 2:**

ANIMAL CRACKERS: Write a function takes a two-word string and returns True if both words begin with same letter animal\_crackers('Levelheaded Llama') --> True animal\_crackers('Crazy Kangaroo') --> False

**Problem 3:**

MAKES TWENTY: Given two integers, return True if the sum of the integers is 20 *or* if one of the integers is 20. If not, return False

makes\_twenty(20,10) --> True

makes\_twenty(12,8) --> True

makes\_twenty(2,3) --> False

**LEVEL 1 PROBLEMS**

**Problem 4:**

OLD MACDONALD: Write a function that capitalizes the first and fourth letters of a name

old\_macdonald('macdonald') --> MacDonald

Note: 'macdonald'.capitalize() returns 'Macdonald'

**Problem 5:**

ALMOST THERE: Given an integer n, return True if n is within 10 of either 100 or 200

almost\_there(90) --> True

almost\_there(104) --> True

almost\_there(150) --> False

almost\_there(209) --> True

NOTE: abs(num) returns the absolute value of a number

**LEVEL 2 PROBLEMS**

**Problem 6:**

Given a list of ints, return True if the array contains a 3 next to a 3 somewhere.

has\_33([1, 3, 3]) → True

has\_33([1, 3, 1, 3]) → False

has\_33([3, 1, 3]) → False

**Problem 7:**

PAPER DOLL: Given a string, return a string where for every character in the original there are three characters paper\_doll('Hello') --> 'HHHeeellllllooo' paper\_doll('Mississippi') --> 'MMMiiissssssiiippppppiii'

**Problem 8:**

BLACKJACK: Given three integers between 1 and 11, if their sum is less than or equal to 21, return their sum. If their sum exceeds 21 *and* there's an eleven, reduce the total sum by 10. Finally, if the sum (even after adjustment) exceeds 21, return 'BUST' blackjack(5,6,7) --> 18 blackjack(9,9,9) --> 'BUST' blackjack(9,9,11) --> 19

**Problem 9:**

SUMMER OF '69: Return the sum of the numbers in the array, except ignore sections of numbers starting with a 6 and extending to the next 9 (every 6 will be followed by at least one 9). Return 0 for no numbers.

summer\_69([1, 3, 5]) --> 9

summer\_69([4, 5, 6, 7, 8, 9]) --> 9

summer\_69([2, 1, 6, 9, 11]) --> 14

**CHALLENGING PROBLEMS**

**Problem 10:**

SPY GAME: Write a function that takes in a list of integers and returns True if it contains 007 in order

spy\_game([1,2,4,0,0,7,5]) --> True

spy\_game([1,0,2,4,0,5,7]) --> True

spy\_game([1,7,2,0,4,5,0]) --> False

**Problem 11:**

COUNT PRIMES: Write a function that returns the *number* of prime numbers that exist up to and including a given number count\_primes(100) --> 25

By convention, 0 and 1 are not prime.

**Problem 12:**

PRINT BIG: Write a function that takes in a single letter, and returns a 5x5 representation of that letter print\_big('a')

out: \*

\* \*

\*\*\*\*\*

\* \*

\* \*

HINT: Consider making a dictionary of possible patterns, and mapping the alphabet to specific 5-line combinations of patterns.  
For purposes of this exercise, it's ok if your dictionary stops at "E".

Problem 13: **Write a Python function that accepts a string and calculates the number of upper case letters and lower case letters.**

Sample String : 'Hello Mr. Rogers, how are you this fine Tuesday?'

Expected Output :

No. of Upper case characters : 4

No. of Lower case Characters : 33

Problem 14: **Write a Python function that takes a list and returns a new list with unique elements of the first list.**

Sample List : [1,1,1,1,2,2,3,3,3,3,4,5]

Unique List : [1, 2, 3, 4, 5]

Problem 14: **Write a Python function to multiply all the numbers in a list.**

Problem 15: **Write a Python function to check whether a string is pangram or not.**

Note : Pangrams are words or sentences containing every letter of the alphabet at least once.

For example : "The quick brown fox jumps over the lazy dog"