

# Problem 141: Confusing Conversions

Difficulty: Medium

Author: Ryan Regensburger, Huntsville, Alabama, United States

Originally Published: Code Quest 2021

## Problem Background

A useful feature in many programming languages is the ability to create functions; sections of code that can be “called” on demand to perform a specific task, avoiding the need to rewrite the same algorithms over and over again. Functions are often given a set of “arguments,” a collection of data presented in a specific order, which the function uses during its execution. Most functions “return” a value, which can be saved to a variable for later use or passed to another function as an argument. Collections of similar or related functions called libraries form the basis of many programming languages and allow developers to create powerful programs much more quickly.

## Problem Description

Lockheed Martin’s Enterprise Operations division is working to develop a set of libraries containing common utility functions, to be shared by all projects across the Lockheed Martin corporation. Your team has been assigned the library focused on data conversions. The quality control team has already created a series of tests to run against your new library; your job is simply to create the functions necessary to pass those tests. The functions you need to create include:

- formatHeight
  - Arguments:
    - X - an integer representing the number of feet
    - Y - an integer representing the number of inches
  - Returns a text string in the format X’Y” (the number of feet, an apostrophe, the number of inches, and a quote)
- formatDate
  - Arguments:
    - X - an integer representing the year
    - Y - an integer representing the month
    - Z - an integer representing the day of the month
  - Returns a text string containing the given date in YYYYMMDD format
- concatenate
  - Arguments:
    - X... - One or more text strings
  - Returns a text string containing all arguments in the order presented, with each argument separated from the next with a comma

## Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include a single line containing the name of a function listed above, followed by the arguments for that function. All values will be separated by spaces.

```
4
formatHeight 5 8
formatDate 2020 5 2
concatenate These are all arguments
concatenate Text can be 123 numbers too
```

## Sample Output

For each test case, your program must execute the named function with the provided arguments and print its return value on a single line.

```
5'8"
20200502
These,are,all,arguments
Text,can,be,123,numbers,too
```