

## CSCI 4041, Fall 2018, Programming Assignment 5

Due Tuesday, 10/9/18, 10:30 AM (submission link on Canvas)

This is not a collaborative assignment; you must design, implement and test the solution(s) on your own. You may not consult or discuss the solution with anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class. Obtaining or sharing solutions to any programming assignment for this class is considered academic misconduct. If you are not sure what this means, consult the class syllabus or discuss it with the course instructor.

You are part of a shadow organization bent on world domination. Your only point of contact is the local restaurant, The Chicken Chancellor; you send in reports by purchasing very specific quantities of chicken nuggets (thankfully, you're not required to actually eat them), and receive your instructions encoded in the numbers on the receipt.

The problem is that being a member of a shadow organization doesn't pay very well, and The Chicken Chancellor does charge you for all those chicken nuggets you ordered. This isn't made any easier by their absurd pricing policies: they allow orders of any positive integer number of chicken nuggets, but the progression of prices doesn't follow any reasonable pattern, and it changes every day.

However, you have a plan to save money: the "reports" you send are based only on the total number of chicken nuggets you order on a given day, so you can split that order up in any number of ways: if you need to send the "infiltration successful" message (3 chicken nuggets), you can do this by purchasing a single order of 3 nuggets, one order of 2 and one order of 1, or three orders of 1 nugget. Now you just need a way to quickly compute the optimal set of orders given the day's pricing structure and the total number of nuggets needed.

Download the PA5.py template from the course website. The template contains a function `optimizeNuggets` that takes as input a list representing the pricing structure for the day and the number of nuggets you need to order, and outputs the cheapest possible set of orders that add up to the required number. The file also includes a few test cases. You'll need to implement an algorithm very similar to rod cutting from Chapter 15 for this to work.

### Requirements:

- You must download the template file PA5.py and edit the `optimizeNuggets` function. Do not edit any other part of the file.
- Your program must run without errors on the version of Python installed on the CSELabs machines, Python 3.5.2. (if you're testing this on CSELabs, you need to type `python3` or `idle3` instead of `python` or `idle` to start the correct version from the terminal)
- You are not allowed to use any Python function that asks for user input, such as `input()`, as this will break the grading script.

- You must implement an algorithm similar in nature to the rod cutting algorithm from Section 15.1 in the textbook. Do not attempt to brute-force every possible combination of nugget orders that add up to the desired quantity: this will not go well for large order sizes.
- However, note that while the textbook algorithm describes how to split a quantity to maximize cost, for this algorithm you want to split your total quantity in a way that minimizes cost.
- This assignment will be graded automatically based on the number of test cases your program passes. There will be several secret test cases in addition to the ones included in the template to ensure you're not hard-coding in the solutions.
- Similar to the previous homework, this program will only run test cases until you fail one, avoiding the problem of having to scroll through test output to find the one broken test case.
- The grading breakdown for this assignment is as follows:
  - 30%: File runs without syntax errors
  - 70%: Passing test cases without breaking any requirements.
- The unedited template file already runs without syntax errors (it does encounter some runtime errors on a few test cases though). This means that if your program causes syntax errors, you will get a better score by just submitting the original template unedited.
- Submit your edited PA5.py file to the Programming Assignment 5 link on Canvas before 10:30 AM on 10/9/18. No credit will be given for late submissions.