

ACM & WiCS Fall 2020 Programming Contest

November 14th, 2020

- Contest URL: <https://domjudge.cs.fsu.edu>
- You have 5 hours to answer questions.
- You may submit solutions in the following languages:
 - C/C++ (1999, 2011)
 - Java 8
 - Python (2.7.13 or 3.5.3)
 - Perl 5.24.1
 - Javascript (Node.js v8.11.0)
- You are only allowed access to official language documentation and COP3014/COP3363 reference material. You are restricted to:
 - C/C++: <http://www.cplusplus.com/reference/>
 - Java: <http://docs.oracle.com/javase/8/docs/api/>
 - Python 2.7.13: <https://docs.python.org/2/>
 - Python 3.5.3: <https://docs.python.org/3/>
 - Perl: <http://perldoc.perl.org/>
 - Javascript:
 - * <https://nodejs.org/docs/latest-v4.x/api/>
 - * <http://developer.mozilla.org/en-US/docs/Web/Javascript/Reference>
 - COP3014/COP3363 Reference:
 - * <https://www.cs.fsu.edu/~vastola/cop3014/>
 - * <https://www.cs.fsu.edu/~vastola/cop3363/>
 - * <https://www.cs.fsu.edu/~jayarama/prog1.html>
- You are also allowed one textbook or material no larger than 8.5" x 11" x 2" volume.
- No other resources (e.g. Stack Overflow, Google, Wikipedia) are permitted. Using non-permitted materials will lead to disqualification.
- Teams are restricted to using one workstation (computer) each.
- Use of a cell phone to circumvent these restrictions will lead to disqualification. Use of cell phones in contest rooms is not permitted.
- The Clarifications tab on Domjudge may be used to submit questions pertaining to each problem. Do not use this feature to request troubleshooting help.

- **All input is redirected via STDIN.**
- **All output must be formatted to specification in terms of capitalization and spacing. Please refer to the example output for each question.**
- Do not include a shebang in your submissions.
- **Scoring:**
 - Teams are ranked according to score. A higher score is rewarded by answering more questions while acquiring fewer penalties.
 - The team that solves the greatest number of questions in the quickest time wins.
 - Teams which solve the same number of problems are ranked by least total time.
 - Teams may resubmit solutions as many times as needed, but incorrect submission attempts will result in time penalties (and thus a lower score.)
 - The scoreboard may be accessed during the first four hours of the contest. The scoreboard will freeze during the final hour.

1 Monster Mash

It's a monster mash! An absolute graveyard smash! All kinds of monsters – from werewolves to vampires and witches to ghosts – will be attending! There's only one problem...how much food do we need!?!?! These hungry monsters aren't going to be too happy without good food. Look the wrong way and they may start nibbling on our human guests. Oh my! Will you help us plan our party?

There will be 4 types of food: Spooky Cake, Pumpkin Pie, Caramel Apple, Candy Corn. Also, we are borderline broke... So, we want to order food that will feed as many guests as possible at a time! This means that foods that can feed more guests have a higher preference. The number of guests that each food type can feed is listed below:

- Spooky Cake - 9 guests
- Pumpkin Pie - 5 guests
- Caramel Apple - 2 guests
- Candy Corn - 1 guest

1.1 Input

A single integer N where $N \leq 100$, that represents the number of guests that will be attending the party.

1.2 Output

The output should be a list of the foods and how many of each needs to be ordered in the sequence of: Spooky Cake, Pumpkin Pie, Caramel Apple, Candy Corn.

Each food item should be formatted in the following way:

`<Food>:<single white-space><number>`

Refer to the sample I/O below, and be sure each food item is Capitalized as well.

1.3 Sample Input/Output

Sample Input 1	Sample output 1
60	Cake: 6 Pie: 1 Apple: 0 Corn: 1
Sample input 2	Sample Output 2
26	Cake: 2 Pie: 1 Apple: 1 Corn: 1