TeamsCode Fall 2020 Online Contest Logistics
Saturday November 21st, 1:00 PM - 5:00 PM Pacific Time

Contest Schedule
❏ 1:00-1:30 - Opening Ceremony and review of rules
❏ 1:30-4:30 - Coding time! Last submission by 4:30 PM
❏ 4:30-5:00 - Raffle and top-scoring winners announced + Closing ceremony

General Rules
❏ All contestants must be a middle or high school student (rising 6th - grad senior)
❏ Teams may not receive any help from anyone outside of their team
❏ Teams may use multiple computers and submit answers in multiple languages
❏ Pre-written code and online reference guides are allowed
❏ There are **two divisions: Intermediate** and **Advanced**. The intermediate division is intended for students with less than a year of coding experience, and the advanced division is intended for everyone who has taken at least one programming class. The registration and problems are **different for each division**. The link to the intermediate contest is teamscode-context.xjoi.net/fall2020int, and the link to the advanced contest is teamscode-context.xjoi.net/fall2020adv.

Problem Format
❏ Description: an overview of the problem and its backstory.
❏ Input Format: specifies how the input will be formatted, including constraints on the size of parameters.
❏ Output Format: specifies how the output should be formatted - if you don't follow this format exactly, your answer will most likely be marked as incorrect.
❏ Sample Input: provides a sample input to help you test your code.
❏ Sample Output: provides the expected output to the sample input.

Submitting Solutions
❏ Solutions will be submitted through the contest page listed above. The code for each problem should be copy-pasted into the box that appears after clicking "Submit Code".
❏ The file size containing your code must not exceed 50 KB.
❏ Your program must run in under 3,000 ms (3 seconds)
❏ The program's memory must not exceed 256 MB

❏ Use **Standard Input** in your code. This means that test cases are directly typed into the console. Here's what standard input is for each of the allowed languages:
Java: `Scanner(System.in)`   **C++:** `cin >>`
Python: `input()`   **C:** `scanf();`
Pascal: `readLn()`

❏ Use **Standard Output** in your code. This means that the output directly prints to the console. Here's what standard output looks like for each of the languages:
Java: `System.out.println()` **C++:** `cout <<`
Python: `print()`   **C:** `printf();`
Pascal: `writeLn()`

Scoring

❏ There are 15 total problems: 5 easy, 5 medium, and 5 hard problems
❏ Each problem has 3 test cases. Easy test cases are worth **10 points** each, Medium test cases are worth **20 points** each, and Hard test cases are worth **30 points** each - Maximum of 900 points if all problems are correct
❏ Output must match **exactly** with expected output to receive points for the test case - there is no partial credit.
❏ Ties will be broken by the timestamp on the last correctly submitted problem.

Prizes

❏ The top 3 scoring teams from each division will win amazon gift cards. The exact amounts each team will win is listed here:
Advanced   **| 1st:** $50 each **| 2nd:** $25 each **| 3rd:** $15 each
Intermediate | 1st: $25 each **| 2nd:** $15 each **| 3rd:** $10 each
❏ Everyone in your team will earn one raffle ticket once you submit your first solution. One additional raffle ticket will be added for anyone who referred others to the contest during registration.
❏ Raffle prizes will be drawn in the last half hour of the contest using a random number generator - we will email those selected for shipment details immediately following the contest.
❏ Raffle prize list: 10.2 inch iPad, Polaroid Now Camera, Active Noise-Cancelling Bluetooth Headphones, Bluetooth Speaker, Corsair Gaming Keyboard, Wireless Mouse, Desk Lamp, 3 Coding T-Shirts

Still have questions? Please don't hesitate to email **contact@teamscode.com** or chat on the YouTube livestream to get them answered!

Example Problem

You are facing off against a golem. In order to beat it, you must cast a magic spell. You are given two numbers. Your task is to find the largest number possible either by adding or multiplying the two numbers together in order to cast the strongest spell.

Input:

The first line contains an integer L. The following L lines will each contain two numbers, N_1 and N_2.

Output:

For each set of two numbers, print the largest number possible through either adding or multiplying the two numbers together.

Sample Input:

```
5
6 12
1 54
2 2
9 -17
-7 -7
```

Sample Output:

```
72
55
4
-8
49
```