COMP 321

WELCOME TO PROGRAMMING CHALLENGES
Instructor

- David Becerra, Trottier Building 3150.
- Website: cs.mcgill.ca/~dbecer/courses/Winter2017/comp321.html
- Email: david.becerraromero@mail.mcgill.ca
- Office Hours: Friday 2PM-3PM (TR 3110) + any other time by email.
Course Goals

• Give students the opportunity to test their algorithm design and programming skills on tricky problems and puzzles.
• Encourage students to join the McGill team for the (world-wide yearly) ACM programming contest.
• To have FUN!.
Right for you if:

- you are looking for a fun course that put a fresh face on standard topics in programming and algorithms,
- you are planning to apply soon to a job interview in a big company,
- you are looking for a 1-credit course,
- you are interested to join the McGill team to participate in the annual ACM International Collegiate Programming Competition
- you are simply motivated by the thrill of competition.
The contest
ICPC

- Premiere global programming competition conducted by and for the world’s universities.
- The oldest, largest, and most prestigious programming contest in the world.
- Aside programming, tests *problem-solving*, team-work, design/decomposition, creativity and testing/debugging
- Sponsored by IBM with ACM support.
- Continued growth in participation:
  - 2016: 40,266 contestants, 2736 universities, 102 countries
ICPC

ACM-ICPC Student Participation

- 1997: 2,520
- 1998: 3,114
- 1999: 4,368
- 2000: 5,904
- 2001: 6,480
- 2002: 7,086
- 2003: 8,640
- 2004: 9,450
- 2005: 12,327
- 2006: 16,818
- 2007: 18,297
- 2008: 21,327
- 2009: 21,957
- 2010: 23,811
- 2011: 25,016
- 2012: 29,479
- 2013: 32,043
- 2014: 38,160
- 2015: 40,266
ICPC – Competition Format

- Runs 5 hours
- Ten or more problems of varying difficulty.
- Teams of 3 students
- Teams share a single PC
- Available languages: C, C++, C#, Java, Python
- Submissions assessed only on correctness
ICPC – Competition Format

• Feedback about a submission is limited: essentially yes, no, run-time error, or time-limit exceeded
• Judges’ secret test data is relentlessly strict
• Incorrect submissions are assigned a time penalty (if the problem is later solved correctly)
• The winner is the team that solves the most problems in the least cumulative time
• Teams have access to a scoreboard and (optionally) visual clues (balloons) to ascertain relative progress
# Results 2016

<table>
<thead>
<tr>
<th>Place</th>
<th>Name</th>
<th>Solved</th>
<th>Time</th>
<th>Last solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>St. Petersburg State University</td>
<td>11</td>
<td>1560</td>
<td>290</td>
</tr>
<tr>
<td>2</td>
<td>Shanghai Jiao Tong University</td>
<td>11</td>
<td>1567</td>
<td>272</td>
</tr>
<tr>
<td>3</td>
<td>Harvard University</td>
<td>10</td>
<td>1358</td>
<td>269</td>
</tr>
<tr>
<td>4</td>
<td>Moscow Institute of Physics &amp; Technology</td>
<td>10</td>
<td>1437</td>
<td>281</td>
</tr>
<tr>
<td>5</td>
<td>University of Warsaw</td>
<td>10</td>
<td>1586</td>
<td>278</td>
</tr>
<tr>
<td>6</td>
<td>Massachusetts Institute of Technology</td>
<td>9</td>
<td>1021</td>
<td>247</td>
</tr>
<tr>
<td>7</td>
<td>St. Petersburg ITMO University</td>
<td>9</td>
<td>1026</td>
<td>208</td>
</tr>
<tr>
<td>8</td>
<td>Ural Federal University</td>
<td>9</td>
<td>1167</td>
<td>212</td>
</tr>
<tr>
<td>9</td>
<td>University of Wroclaw</td>
<td>9</td>
<td>1193</td>
<td>252</td>
</tr>
<tr>
<td>10</td>
<td>Nizhny Novgorod State University</td>
<td>9</td>
<td>1222</td>
<td>292</td>
</tr>
<tr>
<td>11</td>
<td>Lviv National University</td>
<td>9</td>
<td>1430</td>
<td>258</td>
</tr>
<tr>
<td>12</td>
<td>Fudan University</td>
<td>9</td>
<td>1613</td>
<td>289</td>
</tr>
<tr>
<td>13</td>
<td>University of Waterloo</td>
<td>9</td>
<td>1789</td>
<td>276</td>
</tr>
</tbody>
</table>
ICPC - Requirements

Yes

Willing and able to compete in World Finals?

No

Ineligible

Yes

Enrolled in degree program at sponsoring institution?

No

Ineligible

Yes

taking at least 1/2 load OR co-op, exchange, or intern student

Yes

Ineligible

No

Ineligible

Yes

competes for only one institution this year

No

Ineligible

Yes

foreign student visa permits return from World Finals to institution

No

Ineligible

Yes

(number of finals as contestant) >= 2

Ineligible

Return student visa must be secured within 30 days of completion of Regionals.
ICPC - Requirements

Start Period of Eligibility Check

Yes

(year first began post-secondary studies) >= 2012

No

Yes

Born in 1993 or later?

No

Yes

Have completed more than equiv of 8 semesters of full-time study

No

Ineligible

Eligible

Ineligible

No

CONTESTANT MEETS BASIC REQUIREMENTS!

Yes

Ineligible

No

(number of regionals as contestant) >= 5
ICPC - Requirements

No

Coach petitions for extension of eligibility. Provides full history of student academic and work history

Routinely Accepted after Verification
Send to manager@icpc.baylor.edu

ICPC Manager Reviews

Unverified

Ineligible

Verified

Eligible
Levels of Competition

- Local Contest -> Springfield NY, October 15th. (2 McGill teams)
- Regional Contest -> Rochester NY, November 19th. (1 McGill team)
- World Finals -> Rapid City, South Dakota, May 20 – 25

COMP 321 - Information

• 1 credit course

• Pre-Requisites:
  • COMP-203, or COMP-250, or COMP-206
  • MATH-240 and MATH-223
  • Or eagerness to learn (consent of instructor)

• Course hours:
  • Tuesdays: 10:00 – 11:00
  • Labs / Contests: 3 hours – Tuesday from 8 am to 11 am?
COMP 321 - Format

- The course consists of nine 1-hour lectures, and four 3-hour lab meeting / programming contests.
- The lectures are short overviews/reviews of the theory needed to solve certain types of problems (+ hints + tricks + advices).
  - L2 = Data Structures.
  - L3 = Sorting + Strings.
  - L4 + L5 = Divide and Conquer + DP + Greedy.
  - L6 = Graph Theory.
  - L7 = Computational Geometry
  - L8 = Algebra + + Number Theory.
  - L9 = Combinatorics.
- Of course there is not enough time to cover each area in detail. The students are expected to review the areas that they do not master on their own.
COMP 321 - Format

- The lab sessions allow the students to apply their problem solving skills on a set of problems and puzzles using either C, C++, Java or Python.
- One computer + Notebook.
  - Lab #0 (L2 ..... L6) (Individual -> Selection CCSCNE)
  - Lab #2 (L2 ..... L9) (Individual -> Evaluation)
  - Lab #3 (L2 ..... L9) (Team -> Evaluation)
  - Lab #4 (L2 ..... L9) (Team -> Evaluation)
COMP 321 - Format

The lab sessions will consist of 5-10 problems of varyingly difficulty.

FOR EACH PROBLEM
1. Read the problem statement
   – Check the input/output specification!
2. Make the problem abstract
3. Design an algorithm
   – Often the hardest step
4. Implement and debug
5. Submit
6. AC!
   – If not, go back to 4
A friend of yours who is working as a waiter has a problem. A group of xkcd-fans have started to come to the restaurant and order food as in the comic strip below. Each order takes him a lot of time to figure out, but maybe you can help him.

![Comic strip](xkcd.com/287)

**Figure 1: Comic strip [xkcd.com/287](https://xkcd.com/287).**

**Task**

You are to write a program that finds out what was ordered given the total cost of the order and the cost of each item on the menu.
COMP 321 – problem example

Input

The input starts with a line containing one integer \( n \) \((1 \leq n \leq 100)\), the number of items on the menu. The next line contains \( n \) space-separated positive integers \( c_1, c_2, \ldots, c_n \), denoting the cost of each item on the menu in Swedish kronor. No item costs more than 1 000 SEK.

This is followed by a line containing \( m \) \((1 \leq m \leq 1000)\), the number of orders placed, and a line with \( m \) orders. Each order is given as an integer \( s \) \((1 \leq s \leq 30 000)\), the total cost of all ordered items in SEK.

Output

For each order in the input output one line as follows. If there is one unique order giving the specified total cost, output a space-separated list of the numbers of the items on that order in ascending order. If the order contains more than one of the same item, print the corresponding number the appropriate number of times. The first item on the menu has number 1, the second 2, and so on.

If there doesn’t exist an order that gives the specified sum, output Impossible. If there are more than one order that gives the specified sum, output Ambiguous.

Sample Input 1

```
3
4 5 8
3
11 13 14
```

Sample Output 1

```
Impossible
Ambiguous
1 2 2
```
COMP 321 – problem example
Restaurant Orders

Problem statistics

- Submissions: 709
- Accepted submissions: 169
- Submission ratio: 24%
- Authors: 155
- Accepted authors: 120
- Author ratio: 77%

Wrong Answer: 44%

Problem ID: orders
CPU Time limit: 2 seconds
Memory limit: 1024 MB
Difficulty: 5.0
Download:
COMP 321 – Composition of a Team

• Coder:
  • Excellent algorithmic, language and typing skills.

• Algorist:
  • The one best at cracking the problem and sketching out solutions.

• Debugger:
  • One who work off-line from printouts to fix things.

• Of course, these roles change during the flow of the contest.
COMP 321 – Allowed Material

- No internet.
- TEAM NOTEBOOK.
  - Reference material of 20 printed pages or less, to bring with you to the competition floor.
COMP 321 - Tactics

• Know your limitations
  • Identify the easiest problems because you only get credits for correct solutions.

• Keep an eye on the competition
  • View the current standings.

• Avoid wrong answers
  • Correctness is much more important than speed.

• Keep the machine busy
  • Always use the keyboard.

• Clean Debugging
  • Clean printouts.
Grading

• Your final grade will be calculated as follows:
  • 4% for the solution of a training problem.
  • 36% for 12 assignments. (9 home training problems + 3 solutions to contests) (3% each). Each assignment consists of solving 1 or 2 problems.
  • 60% for 3 contests. (20% each).
## COMP 321 Schedule

<table>
<thead>
<tr>
<th>Activity</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture 1</td>
<td>Jan 17</td>
<td>Presentation of the course</td>
</tr>
<tr>
<td>Lecture 2</td>
<td>Jan 24</td>
<td>Data Structures</td>
</tr>
<tr>
<td>Lecture 3</td>
<td>Jan 31</td>
<td>Sorting + String</td>
</tr>
<tr>
<td>Lecture 4</td>
<td>Feb 07</td>
<td>Algorithm Design Paradigms 1</td>
</tr>
<tr>
<td>Lecture 5</td>
<td>Feb 14</td>
<td>Algorithm Design Paradigms 2</td>
</tr>
<tr>
<td>Lecture 6</td>
<td>Feb 21</td>
<td>Graph Theory</td>
</tr>
<tr>
<td>Contest 0</td>
<td>Feb 28, 8:00am-11:00am in ENGTR 3120</td>
<td></td>
</tr>
<tr>
<td>Lecture 7</td>
<td>Mar 07</td>
<td>Computational Geometry</td>
</tr>
<tr>
<td>Lecture 8</td>
<td>Mar 14</td>
<td>Algebra + Number Theory</td>
</tr>
<tr>
<td>Lecture 9</td>
<td>Mar 21</td>
<td>Combinatorics</td>
</tr>
<tr>
<td>Contest 1</td>
<td>Mar 28, 8:00am-11:00am in ENGTR 3120</td>
<td></td>
</tr>
<tr>
<td>Contest 2</td>
<td>Apr 04, 8:00am-11:00am in ENGTR 3120</td>
<td></td>
</tr>
<tr>
<td>CCSCNE Conference</td>
<td>Apr 7 - 8</td>
<td></td>
</tr>
<tr>
<td>Contest 3</td>
<td>Apr 11, 8:00am-11:00am in ENGTR 3120</td>
<td></td>
</tr>
</tbody>
</table>
Training - Algorithm

MONDAY: ‘Student’ attends the class (or contest). (1 - 3 hours)
WEEK DAYS: ‘Student’ reviews the areas that he/she does not master. (2 hours)
WEEK DAYS: ‘Student’ updates his notebook. (1/2 hour)
WEEK DAYS: ‘Student’ solves the proposed training problems and/or perform a contest post-mortem (3 hours).
Course Material

• BOOKS
  • Any other good textbook on algorithms and C++, Java reference

• ONLINE CONTESTS
  • TopCoder
  • Google Code Jam
  • CodeChef
Course Material

- TRAINING PROGRAMS
  - USACO

- ONLINE JUDGES
  - https://open.kattis.com/
  - http://www.spoj.com/
  - https://uva.onlinejudge.org/

- COLLEGIATE CONTESTS
  - https://icpc.baylor.edu/
POLICIES

• Policy on discussion.
  • Respect for the ideas.
  • No discussion = boring class.

• Policy on collaborations.
  • Cheating in this course is a particularly easy (but stupid) thing to do.
    • Discuss with your team.
    • Discuss with other teams.
    • Do not copy code.

• Policy on grading.
  • I am not grading (the robot does that for me)
  • I really care about the learning process, not the marks.
Programming competition Club

- An introductory talk about programming contests.
- Regular meetings to train (solve specific problems together).
- Be mentored by other students.
- Meet friends.
- Have fun!!!
- Planning the season.
time for questions
Questions for you

• How many have experience in programming contests?
• What role will you play in a contest?
  • Coder
  • Algorist
  • Debugger
• How many have taken?
  • COMP-250.
  • COMP-251.
• What is the language you are intending to use?
  • C
  • C++
  • Java.
  • Python.
Questions for you

• How many are eligible to represent McGill?
• How many need a coach petition?
• How many need US visa?
• Can we do competitions on tuesdays morning?