The CENTRE for EDUCATION in MATHEMATICS and COMPUTING

2013 Canadian Computing Competition: Senior Division

Sponsors:

WATERLOO MATHEMATICS

5 YRS 1963-2013

Enriching Mathematics and Computer Science for 50 years

Canadian Computing Competition
Student Instructions for the Senior Problems

1. You may only compete in one competition. If you wish to write the Junior paper, see the other problem set.
2. Be sure to indicate on your **Student Information Form** that you are competing in the **Senior** competition.
3. You have three (3) hours to complete this competition.
4. You should assume that:
   - if your supervising teacher is grading your solutions, all input is from a file named `sX.in`, where `X` is the problem number ($1 \leq \texttt{X} \leq 5$);
   - if you are using the On-line CCC grader, all input is from standard input;
   - all output is to standard output (i.e., to the screen).

   There is no need for prompting. Be sure your output matches the expected output in terms of order, spacing, etc. IT MUST MATCH EXACTLY!
5. Do your own work. Cheating will be dealt with harshly.
6. Do not use any features that the judge (your teacher) will not be able to use while evaluating your programs.
7. Books and written materials are allowed. Any machine-readable materials (like other programs which you have written) are *not* allowed. However, you are allowed to use "standard" libraries for your programming languages; for example, the STL for C++, java.util.*, java.io.*, etc. for Java, and so on.
8. Applications other than editors, compilers, debuggers or other standard programming tools are **not** allowed. Any use of other applications will lead to disqualification.
9. Please use file names that are unique to each problem: for example, please use `s1.pas` or `s1.c` or `s1.java` (or some other appropriate extension) for Problem S1. This will make the evaluator's task a little easier.
10. Your program will be run against test cases other than the sample ones. Be sure you test your program on other test cases. Inefficient solutions may lose marks for some problems, especially Problems 4 and 5. Be sure your code is as efficient (in terms of time) as possible.
11. The Senior problems be given one minute of execution time per test case on a Pentium-4 class computer running at 2GHz (if graded in schools), and similarly, one minute of execution time on the on-line grader (if graded using the on-line grader). If this time limit is exceeded, no points will be awarded for that test case.

12. Note that the top 2 Senior competitors in each region of the country will get a plaque and \$100, and the schools of these competitors will also get a plaque. The regions are:

    - West (BC to Manitoba),
    - Ontario North and East,
    - Metro Toronto area,
    - Ontario Central and West,
    - Quebec and Atlantic.

13. If you finish in the top 20 competitors on this competition, you will be invited to participate in CCC Stage 2, held at the University of Waterloo in May 2013. We will select the Canadian International Olympiad in Informatics (IOI) team from among the top contestants at Stage 2. You should note that IOI 2013 will be held in Australia. Note that you will need to know C, C++ or Pascal if you are invited to Stage 2. But first, do well on this contest!

14. Check the CCC website at the end of March to see how you did on this contest, and to see who the prize winners are. The CCC website is:

    `www.cemc.uwaterloo.ca/ccc`

Problem S1: From 1987 to 2013

Problem Description
You might be surprised to know that 2013 is the first year since 1987 with distinct digits. The years 2014, 2015, 2016, 2017, 2018, 2019 each have distinct digits. 2012 does not have distinct digits, since the digit 2 is repeated.

Given a year, what is the next year with distinct digits?

Input Specification
The input consists of one integer Y ($0 \leq Y \leq 10000$), representing the starting year.

Output Specification
The output will be the single integer D, which is the next year after Y with distinct digits.

Sample Input 1
```
1987
```

Output for Sample Input 1
```
2013
```

Sample Input 2
```
999
```

Output for Sample Input 2
```
1023
```

Problem S2: Bridge transport

Problem Description

A train of railway cars attempts to cross a bridge. The length of each car is 10m but their weights might be different. The bridge is 40m long (thus can hold 4 train cars at one time). The bridge will crack if the total weight of the cars on it at one time is greater than a certain weight. The cars are numbered starting at 1, going up to N, and they cross the bridge in that order (i.e., 1 immediately followed by 2, which is immediately followed by 3, and so on).

What is the largest number T of railway cars such that the train of cars $1...T$ (in order) can cross the bridge?

Input Specification

The first line of input is the maximum weight W ($1 \leq W \leq 100000$) that the bridge can hold at any particular time. The second line of input is the number N ($1 \leq N \leq 100000$) which is the number of railway cars that we wish to move across the bridge. On each of the next N lines of input, there will be a positive integer w_i ($1 \leq i \leq N$, $1 \leq w_i \leq 100000$) which represents the weight of the ith railway car in the sequence.

Output Specification

Your output should be a non-negative integer representing the maximum number of railway cars that can be brought across the bridge in the order specified.

Sample Input 1

```
100
6
50
30
10
10
40
50
```

Output for Sample Input 1

```
5
```

Explanation of Output for Sample Input 1

The first four railway cars have total weight $50 + 30 + 10 + 10 = 100$, which is not greater than what the bridge can hold. When the first railway car leaves, and the next comes on, we have a total weight of $30 + 10 + 10 + 40 = 90$, which is not greater than what the bridge can hold. The last four cars would cause the bridge to break, since $10 + 10 + 40 + 50 = 110$ which is greater than the bridge can hold. So, only the first 5 railway cars can be taken across the bridge.

Sample Input 2

```
100
3
150
1
1
```

Output for Sample Input 2

```
0
```

Explanation of Output for Sample Input 2

When the first railway car enters the bridge, its weight of 150 will exceed the maximum weight the bridge can hold. Thus, we cannot bring any railway cars across the bridge.

Problem S3: Chances of winning

Problem Description
You want to determine the chances that your favourite team will be the champion of a small tournament.

There are exactly four teams. At the end of the tournament, a total of six games will have been played with each team playing every other team exactly once. For each game, either one team wins (and the other loses), or the game ends in a tie. If the game does not end in a tie, the winning team is awarded three points and the losing team is awarded zero points. If the game ends in a tie, each team is awarded one point.

Your favourite team will only be the champion if it ends the tournament with strictly more total points than every other team (i.e., a tie for first place is not good enough for your favourite team).

The tournament is not over yet but you know the scores of every game that has already been played. You want to consider all possible ways points could be awarded in the remaining games that have not yet been played and determine in how many of these cases your favourite team will be the tournament champion.

Input Specification
The first line of input will contain an integer T which is your favourite team ($1 \leq T \leq 4$).
The second line will contain an integer G, the number of games already played ($0 \leq G \leq 5$).
The next G lines will give the results of games that have already been played. Each of these lines will consist of four integers A, B, S_A, S_B separated by single spaces where $1 \leq A < B \leq 4$, and $S_A, S_B \geq 0$. This corresponds to a game between team A (which had score S_A) and team B (which had score S_B) where team A won if $S_A > S_B$, team B won if $S_A < S_B$ and the game ended in a tie if $S_A = S_B$. The pairs A and B on the input lines are distinct, since no pair of teams plays twice.

Output Specification
The output will consist of a single integer which is the number of times that team T is the champion over all possible awarding of points in the remaining games in the tournament.

Sample Input 1
```
3
3
1 3 7 5
3 4 0 8
2 4 2 2
```

Output for Sample Input 1
```
0
```

Explanation of Output for Sample Input 1
Team 3 has lost two of its three games, and team 4 has tied one and won one, which gives 4 points to team 4. Even if team 3 wins its final game, it cannot have more points than team 4, and thus, will not be champion.

Sample Input 2

```
3
4
1 3 5 7
3 4 8 0
2 4 2 2
1 2 5 5
```

Output for Sample Input 2

```
9
```

Explanation of Output for Sample Input 2
After these games, we know the following:

Team	Points
1	1
2	2
3	6
4	1

There are two remaining games (team 3 plays team 2; team 1 plays team 4). Teams 1, 2 or 4 cannot achieve 6 points, since even if they win their final games, their final point totals will be 4, 5 and 4 respectively. Thus, out of the 9 possible outcomes (2 matches with 3 different possible results per match), team 3 will be the champion in all 9 outcomes.

Problem S4: Who is taller?

Problem Description

You have a few minutes before your class starts, and you decide to compare the heights of your classmates. You don't have an accurate measuring device, so you just compare relative heights between two people: you stand two people back-to-back, and determine which one of the two is taller. Conveniently, none of your classmates are the same height, and you always compare correctly (i.e., you never make a mistake in your comparisons).

After you have done all of your comparisons, you would like to determine who the tallest person is between two particular classmates.

Input Specification

The first line contains two integers N and M separated by one space. N, the number of people in the class, is an integer with $1 \leq N \leq 1000000$. M, the number of comparisons that have already been done, is an integer with $1 \leq M \leq 10000000$. Each of the next M lines contains two distinct integers x and y ($1 \leq x, y \leq N$) separated by a space, indicating that person number x was determined to be taller than person number y. Finally, the last line contains two distinct integers p and q ($1 \leq p, q \leq N$) separated by one space: your goal is to determine, if possible, whether person p is taller than person q. Note that it may be the case that neither p nor q occur in the input concerning measurements between classmates, and each measurement between any two particular people will be recorded exactly once.

Output Specification

The output is one line, containing one of three possible strings:

- `yes` (if p is taller than q),
- `no` (if q is taller than p),
- `unknown` (if there is not enough information to determine the relative heights of p and q).

Sample Input 1

```
10 3
8 4
3 8
4 2
3 2
```

Output for Sample Input 1

```
yes
```

Sample Input 2

```
10 3
3 8
2 8
3 4
3 2
```

Output for Sample Input 2

```
unknown
```

Problem S5: Factor Solitaire

Problem Description
In the game of Factor Solitaire, you start with the number 1, and try to change it to some given target number n by repeatedly using the following operation. In each step, if c is your current number, you split it into two positive factors a, b of your choice such that $c = a * b$. You then add a to your current number c to get your new current number. Doing this costs you b points. You continue doing this until your current number is n, and you try to achieve this at the cost of a minimum total number of points.

For example, here is one way to get to 15:

- start with 1
- change 1 to 1+1 = 2 — cost so far is 1
- change 2 to 2+1 = 3 — cost so far is 1+2
- change 3 to 3+3 = 6 — cost so far is 1+2+1
- change 6 to 6+6 = 12 — cost so far is 1+2+1+1
- change 12 to 12+3 = 15 — done, total cost is 1+2+1+1+4=9.

In fact, this is the minimum possible total cost to get 15. You want to compute the minimum total cost for other target end numbers.

Input Specification
The input consists of a single integer $N \geq 1$. In at least half of the cases $N \leq 50000$, in at least another quarter of the cases $N \leq 500000$, and in the remaining cases $N \leq 5000000$.

Output Specification
Compute the minimum cost that gets you to N.

Sample Input 1
```
15
```

Output for Sample Input 1
```
9
```

Sample Input 2
```
2013
```

Output for Sample Input 2

```
91
```

Explanation of Output for Sample Input 2

For example, start with 1, then get to 2, 4, 5, 10, 15, 30, 60, 61, 122, 244, 305, 610, 671, 1342, and then 2013.