Fourth Annual Quinnipiac University Programming Competition for High School Students

Quinnipiac University

April 5, 2008

Instructions

1. Submit a program by attaching all necessary files to an e-mail to `cscjudge`. Include the number of the problem you are solving in the subject line of your e-mail.

   Questions may also be addressed to `cscjudge`; put the word "Question" in the subject line, along with the number of the question you are asking about if your question pertains to a particular problem.

2. You may assume the input is formatted as shown in the problem, and your program must work on such input. You may assume there are no tabs in the input. In the sample runs of the programs, all input is boldfaced.

3. Your feedback on a submitted problem will be e-mailed in a reply and will be limited to one of the following:

   (a) COMPILE ERROR: The program did not compile

   (b) PROGRAM NOT RUNNABLE: The program compiled, but did not run as a stand-alone program.

   (c) CRASH: The program crashed or threw an exception when it was run

   (d) TIMED OUT: The program did not finish in a reasonable amount of time

   (e) INCORRECT OUTPUT Either the answer is incorrect, or it is not presented in the required form. **Your program should match the sample output *exactly* when given the same input, with no extra spaces, even at the end of the line.**

   (f) ACCEPTED

   If your program crashes, produces incorrect output, or times out, no information will be given as to the input that caused the problem.

1 Day of the Week

Write a program which determines the day of the week for a date, given the day of the week on which January 1 occurs for that year, and whether the year is a leap year or not. The input for this problem consists of three lines. The first line gives the day of the week for January 1 on a line by itself, the second line contains the word "yes" if the year in question is a leap year, or "no" otherwise, and the last line contains the month, represented as an integer between 1 and 12 (inclusive), followed by the date—an integer between 1 and 31 (inclusive). You may assume that the first letter of the first line is uppercase, and the rest of that line is lowercase, and that the words "yes" and "no" are all lowercase. For example, to find the day of the week for April 5, 2008 given that January 1, 2008 was a Tuesday, running the program would generate the following output (the input is boldfaced, as it is for every problem in this competition):

```
Jan. 1: Tuesday
Leap year: yes
Date: 4 5
Saturday
```

For your convenience, the number of days in each month is given by the following table:

Month number	Month name	Days in Month
1	January	31
2	February	28 (29 in leap years)
3	March	31
4	April	30
5	May	31
6	June	30
7	July	31
8	August	31
9	September	30
10	October	31
11	November	30
12	December	31

2 Musical Chairs variation

A group of N children are playing the following variation of Musical Chairs. The children sit in a circle, and count off k spots at a time. Each time, the kth chair is removed from the circle, and the child in that chair is eliminated from the game. The last child sitting is the winner.
If the spots are numbered from 1 to N, the first count starts from spot 1; each subsequent count started with the person immediately after the last person eliminated. After a chair is removed, that position was no longer included in the count.
For this problem, you are given N and k as input, each on a line by itself, and you need to determine the position of the winner. You may assume $2 \leq N \leq 100$ and $1 \leq k \leq N$. For example, if there are 10 children, and every third seat is eliminated, running the program should look as follows.

```
N: 10
k: 3
Spot 4
```

To explain this answer, let's label the children in the game from 1 to 10, and watch how the game proceeds.

Round	**Remaining children**									
0	1	2	3	4	5	6	7	8	9	10
1	1	2	**X**	4	5	**X**	7	8	**X**	10
2	1	**X**	*X*	4	5	*X*	**X**	8	*X*	10
3	**X**	*X*	*X*	4	5	*X*	*X*	**X**	*X*	10
4	*X*	*X*	*X*	4	**X**	*X*	*X*	*X*	*X*	10
5	*X*	*X*	*X*	4	*X*	*X*	*X*	*X*	*X*	**X**

In the first round, the children in spots 3, 6, and 9 are eliminated. In the next round, we start counting with the child in spot 10, so the child in spot 2 is eliminated, followed by the child in spot 7 (remember, the children in spots 3 and 6 were eliminated the round before, so we ignore those spots in subsequent rounds). The rest of the count proceeds as above; in each round, the boldfaced X's represent children eliminated in that round. Finally in the last round, we count the child in spot 10, then the child in spot 4, and then we eliminate the child in spot 10.

3 The Electoral College

In a Presidential election in the United States, each state is given votes in the Electoral College based on its size: each state gets two votes more than the number of its Congressional Districts. For example, Connecticut has five Congressional districts, so it gets seven electoral votes. Most states use a "winner-take-all" (WTA) system to award its electoral votes: whoever wins that state gets all the electoral votes. Some states, however, use a system in which the winner of each Congressional District gets an electoral vote for that district, and the winner of the entire state is awarded two extra votes. For example, if a candidate wins three of Connecticut's five districts, and also wins the state, that candidate would get a total of five electoral votes, but the other candidate would get two electoral votes for the districts he or she won.

For this problem, we assume there are exactly two candidates: a Democrat, and a Republican. Write a program which computes *the total popular vote*, and *the number of electoral votes that would be won in each system* from one or more states. Your program should prompt the user for the number of states. Then for each state, prompt the user for the number of Congressional districts, and for the number of votes won by the Democrat and the number won by the Republican (in that order) in each district. Your program should print out the total number of votes received by each candidate, along the number of electoral votes that would be won in a winner-take-all (WTA) system within each state, and the number of electoral votes won if the voting were done by congressional districts (CD). For example, using two states might give the following output:

```
States: 2
State 1 districts: 4
District 1: 200000 220000
District 2: 240000 120000
District 3: 180000 200000
District 4: 160000 220000

State 2 districts: 1
District 1: 225000 250000

Votes: Dem 1005000 Rep 1010000
WTA: Dem 6 Rep 3
CD: Dem 3 Rep 6
```

In this example, the Democrat wins district 2 in the first state, and the Democrat also wins the first state by a margin of 780,000 to 760,000. This gives the Democrat six electoral votes under the winner-take-all system, but only three electoral votes by Congressional districts. The Republican wins the other three districts in the first state (but gets no electoral votes in the winner-take-all system), and wins all three electoral votes in the second state under either system.

You may assume that there are at most 51 states (including, possibly, Washington D. C.), that each state has at most 52 Congressional districts (the number of districts in California), there are at most 436 districts among all the states, and there are at most $500,000$ voters in each district. You may also assume there are no ties, either within a district, or at the state level.

4 Adding Fractions

Write a program which takes two fractions, each on a line by itself, and prints out the sum as a *mixed number* in lowest terms. A *mixed number* consists of an integer portion and a fraction portion. The sum of the fractions $\frac{a}{b}$ and $\frac{c}{d}$ is given by the formula $\frac{ad+bc}{bd}$. For example, if the two fractions are $\frac{7}{12}$ and $\frac{3}{4}$, the sum is $\frac{7 \cdot 4 + 12 \cdot 3}{12 \cdot 4} = \frac{64}{48} = \frac{4}{3}$ (in lowest terms) $= 1\frac{1}{3}$ (as a mixed number). You may assume that both numerators and both denominators are positive integers between 1 and $10,000$ (inclusive). If the sum is an integer, omit the fractional portion in your output; if the sum is less than 1, omit the integer portion. The fractional portion of your answer must also be presented in lowest terms, i. e. the numerator and denominator should have no common factors. Thus, running the program three times might produce the following output:

```
Fraction 1: 7/12
Fraction 2: 3/4
1 1/3
```

```
Fraction 1: 1/6
Fraction 2: 2/15
3/10
```

```
Fraction 1: 2/3
Fraction 2: 4/3
2
```

The first example is the one given above; the second example gives a case where the result is less than 1, so we omit the integer portion of the answer, and the third example gives a case where we omit the fractional portion of the answer.

5 The Partition Problem

Given a collection of positive integers, the partition problem asks whether we can split those integers into two groups with the same sum, with each integer is in exactly one of the groups. For example, if the original integers are 1, 2, 3, 4, and 8, we could write:

$$2 + 3 + 4 = 1 + 8,$$

but if the numbers are 1, 2, 3, 4, and 5, one sum must be even and the other must be odd, no matter how we split the numbers.

Write a program which prompts the user to specify how many integers are in the collection of integers, and then reads in the integers on a single line. If it is possible to solve the partition problem, your program then prints out the two groups, *beginning with the group with fewer integers, or if both groups have the same number of integers, beginning with the group which contains the first integer entered*. Within each group, the numbers should be printed in the order they were entered. If it is not possible to solve the problem, your program should print the word "Impossible." If there is more than one solution, any solution is acceptable. You may assume that there are between one and ten integers (inclusive), and they are all between 1 and $1,000,000$.

Running the program several times might produce the following output:

```
How many integers? 5
Integers: 1 2 3 4 10
Group 1: 10
Group 2: 1 2 3 4
```

```
How many integers? 5
Integers: 8 4 3 2 1
Group 1: 8 1
Group 2: 4 3 2
```

```
How many integers? 4
Integers: 1 3 4 2
Group 1: 1 4
Group 2: 3 2
```

```
How many integers? 4
Integers: 3 4 1 2
Group 1: 3 2
Group 2:  4 1
```

```
How many integers? 5
Integers: 1 2 3 4 5
Impossible
```

Notice that in the third and fourth examples, when both sets have two integers, the first set printed in each case is the one that includes the first integer entered.

6 Two-card Poker

Bob Cat and his friend Quinn I. Piac have devised a new version of poker. The purpose of this program is to determine which player has the better hand. Each hand consists of two cards from a standard deck of fifty-two playing cards. There are four suits—Clubs, Hearts, Diamonds, and Spades—and each suit has thirteen ranks, the Ace, King, Queen, Jack, Ten, 9, 8, 7, 6, 5, 4, 3, and 2 in that order, though the ace may be either high or low in a straight or a straight flush, described below.
The hands are ranked from highest to lowest (based on which hands are the least likely to occur) as follows:

- A **Straight Flush**: Two cards of the same suit whose ranks are adjacent, such as the Ten of Hearts and the 9 of Hearts. If both players have a straight flush, the player whose top card is higher wins. Thus, the best straight flush (and thus, the best possible hand) consists of an Ace and a King. A 2 and an Ace of the same suit is also a straight flush, but the 2 is the higher card, so this would be the lowest possible straight flush.

- A **Pair**: Two cards of the same rank, such as two Jacks. If both players have a pair, the player with the higher ranking pair wins.

- A **Straight**: Two cards whose ranks are adjacent, but which are of different suits, such as a 7 of clubs and a 6 of diamonds. If both players have straights, ties are broken in the same way as they are broken for a straight flush: the player with the higher ranking top card is the winner.

- A **Flush**: Two cards of the same suit whose ranks are not adjacent. If both players have a flush, the player with the highest ranking card wins, or if both players have the same top card, the player whose other card has the higher ranking wins.

- **High card**: If no one has a pair, a straight, or a flush, the hands are ranked as they would be for a flush: the player with the higher top card wins, or if both players have the same top card, the player with the higher ranking bottom card wins.

Write a program that takes as input two hands and determines which hand is better. The first line of input gives the two cards in Bob's hand, separated by one or more spaces, and the second line gives the cards in Quinn's hand. Each card is represented by two characters: the first gives its rank (using 'A', 'K', 'Q', 'J', and 'T' for the ace, king, queen, jack, and ten, respectively, and '9' through '2' for the other cards) and the second gives the first letter of the suit. Thus, "KS TH" represents a hand with the King of Spades and the Ten of Hearts. You may assume all letters are uppercase.
Your output should be either the name of the winner, or the word "Tie" if neither player wins. Sample output from this program is on the following page.

Sample poker output

Running the program several times might produce the following output:

```
Bob's hand: 2S AS
Quinn's hand: AC AD
Bob
```

```
Bob's hand: JC TC
Quinn's hand: AH 2H
Bob
```

```
Bob's hand: 8H JH
Quinn's hand: 6C 7D
Quinn
```

```
Bob's hand: QH 5S
Quinn's hand: KC 4D
Quinn
```

```
Bob's hand: QC 9D
Quinn's hand: QH 6S
Bob
```

```
Bob's hand: 3S 9D
Quinn's hand: 9H 3D
Tie
```

The first two examples remind us that an ace and a 2 are a straight (or a straight flush), but they are the lowest possible straight flush. The third example shows us that a straight beats a flush. In the fourth and fifth examples, we look at the highest card in each hand because neither player has a straight, a flush, or a pair, and we look at the lower card when both players have the same high card. The last example reminds us that ties are possible as well.