Problem Set

Trinity University ACM
High School Programming Competition
April 9th, 2011

Problem 0 - A Slow Boat To Work

In order to get to work each day, Boudreaux and Thibodeaux use an airboat to travel across the bayou. However, there's one pesky alligator who always hunts in the bayou. The alligator gets hungry every 19 hours, and he goes out to hunt whenever he gets hungry. It can take him different lengths of time to catch his prey. After the catch, the alligator will not be hungry again for another 19 hours, so he will go home. The alligator begins hunting at 6:00 each Monday. Boudreaux and Thibodeaux would like to know if they would be safe in their travels to and from work each week, knowing that they are traveling to work from 8:00 to 9:00 and traveling home from 17:00 to 18:00. You can assume that they arrive at their given locations at exactly 9:00 and 18:00, meaning they are safe starting at those times. They will be safe if the alligator is not hunting while they are traveling across the bayou. The alligator begins hunting exactly 19 hours from when he last caught his prey. When he catches his prey, he returns home immediately (so if he catches his prey at 8:00, Boudreaux and Thibodeaux would be safe). All times are given in military time (on a 24 hour scale, where 0:00 is midnight and 13:00 is 1:00 PM, for example).

Input: The first line of input will be a single integer, N, which indicates the number of input sets to process. Each data set will consist of a single line containing a list of space-separated integers. Each integer is the number of hours it takes the alligator to catch his prey for the first, second, third, etc. time he is hunting, respectively. At the beginning of each data set, the starting hunt time is reset to 6:00 on Monday. The end of the last hunting time will be before Saturday. It never takes the alligator more than 10 hours to catch prey.

Output: The output for each data set will consist of a single line saying either "Safe!" or "Gator Chow" depending on whether Boudreaux and Thibodeaux will be safe during their travels for the week. If the alligator is not out hunting during the times when Boudreaux and Thibodeaux are traveling to work, then they are safe for that day. Boudreaux and Thibodeaux must be safe for all five weekdays in order for you to output "Safe!"

Sample Input:

```
2
3 3 2 4 8 4 5 5
1 4 5 6 1 3
```

Sample Output:

```
Gator Chow
Safe!
```

Problem 1 - Tweeter

Boudreaux recently signed up for a Twitter account. Unfortunately, he hasn’t quite gotten a hang of the whole 160 characters thing. Instead of working on controlling his verbosity, he has decided to write a program that will nicely convert his long ramblings into properly sized tweets.

Input: The input starts with a single line that has the number of input sets. Each input set is a single line of text up to 1000 characters in length. Input lines will not be blank.

Output: Each output begins with a line that says “Output #” where # is the number of the input starting with 1. After that will be several lines of text, one for each tweet. You have to break tweets on spaces from the original message with each tweet having as close to 160 characters as possible (except the last might be short). In other words, you aren’t allowed to break a word across two tweets. Tweets should neither start nor end with spaces. Also, multiple adjacent spaces in the input should be reduced to a single space.

Sample Input:

```
3
This is some text that we will break up into parts for a tweet.  Because tweets are reasonably long that means that this has to be a long line of text. So I get to keep typing for a while. Can you just imagine what the judge's input looks like?
There   are  too     many   spaces  here.
The proctors are laughing at you holding this paper so close to your face right now.
```

Sample Output:

```
Output 1
This is some text that we will break up into parts for a tweet. Because tweets are reasonably long that means that this has to be a long line of text. So I get
to keep typing for a while. Can you just imagine what the judge's input looks like?
Output 2
There are too many spaces here.
Output 3
The proctors are laughing at you holding this paper so close to your face right now.
```

Problem 2 - Shoutcasters Galore!

In the world of competitive gaming, the people who announce and commentate games are called shoutcasters. Boudreaux and Thibodeaux really like Starcraft 2, and they have decided to attend one of the major Starcraft 2 tournaments in their area. Some of their favorite shoutcasters and professional players are going to be there, but unfortunately, the tournament planners did not make good selections when choosing which shoutcasters are casting each game. Boudreaux and Thibodeaux think they can do a better job, and you are going to help them! Each shoutcaster has a list of players whose games he/she likes to cast. Given a list of shoutcasters and their preferred players, as well as a list of match-ups that are occurring simultaneously, you must determine the maximum number of shoutcasters that can be satisfied with their casting assignment. A shoutcaster is satisfied if he/she is casting a game in which at least one of his preferred players is competing.

Input:
The first line of input will be a single integer, N, which indicates the number of input sets to process. Each input sets begins with a line that contains a single integer, M, indicating the number of match-ups. The next M lines will contain the match-ups, in the format "player1 vs player2". After the match-ups will be a line containing a single integer, S, which indicates the number of shoutcasters. S will be greater than or equal to M. The following S lines will each have a list of space-separated names, one word per name. The first name on each of the S lines will be the name of the shoutcaster. The other names on that line will be that shoutcaster's preferred players. You can assume that each of the S lines will contain at least two names - the shoutcaster's name and at least one player's name. All of the shoutcaster's preferred players will be competing in at least one of the match-ups, but not all players will necessarily be represented as a preferred player. Each player will only be listed in one match-up. No shoutcaster will be a fan of more than five players. (1<M<15, 1<S<20)

Output:
For each input set, output a single line containing the maximum number of casters that can be satisfied with their casting assignments, as described above. The output for each input set should be on a separate line.

Sample Input:

```
1
3
IdrA vs Huk
INcontroL vs Tyler
Ret vs Jinro
3
Day[9] Jinro Tyler INcontroL
```

```
Tasteless IdrA
Artosis Huk
```

Sample Output:

```
2
```

Problem 3 - Reptile Revenge

After an unhappy encounter in one of the events described in problem 0, Boudreaux and Thibodeaux decide they need to take matters into their own hands and do some gator hunting. Unfortunately, they aren't particularly good at this activity, and are not always successful in their reptilian pursuits. They often spend too much time running around swamps to stay in their budget. (The copious amounts of mud limits their travel to 3 meters/second.) In an effort to cut spending, they have limited themselves to spending 100 minutes to catch a single alligator. Your job is to determine the **fastest** time to catch the alligator, if possible at all.

Input:
The first line of input will contain the integer N, the number of data sets for this problem. The data sets contain two lines:
The first line will contain the 4 numbers: coordinates of the alligator that Boudreaux and Thibodeaux need to catch (given as X Y), along with its direction expressed as a vector (R S), meaning that the alligator can travel R meters to the right and S meters up in one second (note that R and S can be negative, allowing the alligator to move left or down, as well).
The second line will contain the coordinates of Boudreaux and Thibodeaux.

Output:
The output for each problem set will consist of one line: If they are able to capture the alligator the output should be a single number for how many seconds it took accurate to 0.01 seconds. If not, the output should be NO.

Sample Input:

```
2
2 1 3 5
0 0
5 2 -2 -1
2 2
```

Sample Output:

```
NO
0.6213203435596424
```

(Note the second number could be different as long as it is accurate to 0.01 sec.)

Problem 4 - Cribbage

It is not uncommon that Boudreaux and Thibodeaux find themselves out in the swamp with little to do on the weekends. As a result, they have become fairly avid card players. One of their favorite games in Cribbage. Unfortunately, the scoring for Cribbage is fairly complex and while they like to play the game, they are simply too lazy to do the scoring. They have asked you to help out by writing a program that will do the scoring for a limited rule set.

Rules: Each player is given four cards in his hand. There is one card, the cut card, which both players use to calculate points. The basic premise of the game is to add up points based off the cards that you have in a hand supplemented with the cut card. Each card's number value (note: values are distinct from points) is the number printed on the card, with Face cards having a number value of 10 and Aces having a number value of 1.

Points are gained in the following way:

- Every unique card combination whose value sum is 15 is worth 2 points. E.g. for cards (8c,7d) you would receive two points. (8c,7d) and (7d,8c) are NOT unique combinations.
- Every unique par is worth 2 points. So, 3 of a kind is 3 unique pairs (6 points) and 4 of a kind is 6 unique pairs (worth 12 points)
- A flush of four cards (all of the cards have the same suit) is worth 4 points. Note that this can only happen with the cards in your hand. So if you have 3 cards in your hand that are clubs and the cut card is a club: it is not a valid four card flush.
- A five card flush is worth 5 points. (Where all the cards in your hand are the same suit as well as the cut card being the same suit)
- Points from a four card and five card flush do not stack. I.e. you only get points for a four-flush or a five-flush, not both.

Input:
The first line of input will be a single integer for the number data sets. Each data set contains three lines. The first two lines contain one hand each: first line being Boudreaux's hand and the second being Thibodeaux's hand. Each hand has four playing cards: represented as A,2,3...,10,J,Q,K next to a single character designating the suit: s,h,c,d. (E.g. The 10 of diamonds would be 10d) The third line contains the cut, which is a single card represented the same as above.

Output:
The output for this program should be a single line for each input containing the name of the winner (Boudreaux or Thibodeaux) and the point total for the winning hand, separated by a single space. In case of a tie, you should output TIE then the point total for the hand, separated by a space.

Sample Input:

```
3
As Ah Ac 6d
Kc Qc Jc 10c
4h
4h 6d 8s Kc
Ah 6d 7c 8c
2d
Kc Kd Kh Js
Qh Qs Qc 10d
5h
```

Sample Output:

```
Boudreaux 6
Thibodeaux 6
TIE 14
```

Problem 5 - Diamonds of Boredom

Boudreaux and Thibodeaux are very bored. How bored are they, you ask? They are so bored, that they have decided to write their names together in the shape of a diamond. After writing a few by hand, they have decided they would rather a computer do this tedious task for them. However, as you can probably tell, Boudreaux and Thibodeaux don't know how to write computer programs, so it is once again your job to help them. You are to write a program that prints ASCII diamonds using "Boudreaux and Thibodeaux" as described below.

ASCII diamonds can be drawn with integer side lengths. Each layer of this diamond is usually drawn with a single ASCII alphabet character. In this case, we will use the string "Boudreaux and Thibodeaux" starting with 'B' in the center, an 'o' in the next level, a 'u' after that, etc. After the last x the sequence repeats again.

Input: The first line of input will be a single integer, 1 <= N <= 100, which indicates the number of input sets to process. Each of the input sets will contain a single integer, 1 <= S <= 50 that represents the size of the diamond to draw.

Output: For each set you should print the diamond of the given size.

Sample Input:

```
3
2
12
4
```

Sample Output:

```
 o
oBo
 o
           n
          nan
         na an
        na x an
       na xux an
      na xuaux an
     na xuaeaux an
    na xuaereaux an
   na xuaerdreaux an
  na xuaerdudreaux an
 na xuaerduoudreaux an
na xuaerduoBoudreaux an
 na xuaerduoudreaux an
  na xuaerdudreaux an
   na xuaerdreaux an
```

```
    na xuaereaux an
     na xuaeaux an
      na xuaux an
       na xux an
        na x an
         na an
          nan
           n
   d
  dud
 duoud
duoBoud
 duoud
  dud
   d
```

Problem 6 - Encoded Messaging

Based on their rather interesting method of getting to and from work, you probably wouldn't have guessed that Boudreaux and Thibodeaux are actually secret agents! Because of the heavy emphasis on secrecy in the office, Boudreaux and Thibodeaux must encrypt any messages they send. You are to write a program to help them encode their messages, according to the description below.

The encryption method will be based on a given code word. The number of characters in the code word will determine how many substitution alphabets to generate. Each substitution alphabet will begin with the next letter in the code word, followed by the other letters in the English alphabet, in order, and wrapping around when the letter 'z' is reached. For example, say the code word is "code". The substitution alphabets will be:

```
    cdefghijklmnopqrstuvwxyzab
    opqrstuvwxyzabcdefghijklmn
    defghijklmnopqrstuvwxyzabc
    efghijklmnopqrstuvwxyzabcd
```

All code words will be a single, lowercase word containing the characters 'a' through 'z'. Each given string will then be encoded using these substitution alphabets by first removing all spaces and non-alphabet characters from the string, then converting the string to all lowercase letters (for example, "Hello, World!" should be converted to "helloworld"). Then, each letter in the new string will be substituted using the generated alphabets. The first letter in the string will be substituted using the first alphabet, the second letter using the second alphabet, etc. and wrapping back to the first alphabet when necessary until all letters have been encoded. When substituting letters, the following scheme is used:

a will be substituted with the 1st letter in the substitution alphabet
b will be substituted with the 2nd letter in the substitution alphabet
c will be substituted with the 3rd letter in the substitution alphabet
etc.
z will be substituted with the 26th letter in the substitution alphabet

Input: The input will start with a single line that has the number of input sets to process. Each input set will start with one line containing the code word. The code word will be entirely lowercase letters. After that will be a line with the phrase that must be encoded using the given code word.

Output: For each set, you should print one line with the complete encoded text.

Sample Input:

```
3
```

```
code
Dear mom: School is great. Send money. Love, your son.
boudreaux
Buy dynamite for gators
cajun
I would rather be playing Starcraft
```

Sample Output:

```
fsdvocpwevrsnwvktsdxusqhocqiazrzgmrytgrr
cisgpragfuszrikanlsg
kwxoyfrjnugrkycnahcaiscueerjzg
```

Problem 7 - Cajun Friendship

Boudreaux and Thibodeaux are quite popular. Even though they seem to only ever hang out with each other, they actually have a bunch of friends. However, their friends don't all know each other, and the friends that do know each other don't always get along. Boudreaux and Thibodeaux would like to know how many potential friends each of their friends has, using the adage "the enemy of my enemy is my friend." Given friendship information, you are going to write a program to determine how many "friends" a given person has.

Input: The first line of input will be a single integer, N, which indicates the number of input sets to process. The following line will contain a single integer, R, indicating the number of relationships. The next R lines will contain the relations, which will be a name, followed by R letters indicating the relationships separated by spaces. The relationship letters will be 'F' for Friend, 'E' for Enemy, 'N' for Neutral, or 'S' for Self. The order of the relationship letters corresponds to the order of names given in the R lines (e.g. the third relationship letter is the relationship towards the person given on the third of the R lines). All relationships will be reciprocal, so if person 1 is a friend to person 2, person 2 will also be a friend to person 1. After the R lines will be a single line, containing the name of one of the people in the above relationships.

Output: The output will be a single integer containing the number of "friends" of the person given on the last line of the input. You will calculate this number by counting how many direct friends this person has, as well as how many potential friends he/she has. A potential friend is someone who the person has a neutral relationship with, and who is an enemy of any one or more of his/her enemies. A person is not a friend of himself/herself.

Sample Input:

```
1
4
Bob S F F F
Sue F S E E
Fred F E S N
Joe F E N S
Joe
```

Sample Output:

```
2
```

Problem 8 - Random Battles

Boudreaux and Thibodeaux love to play Starcraft 2. They are planning a night to play against each other to prove, once and for all, who is the better player. Boudreaux prefers to play as Protoss, while Thibodeaux likes Zerg. However, to make things more interesting, they have both agreed to randomly select which race they will be playing for each game. Given the race of each player, your job is to predict who will win that match. If both players are using their preferred race, you will predict a “Tie”. However, if only one player gets his preferred race, you will predict that player to win. If neither player gets his preferred race, you will again predict a “Tie”.

Input:
The first line of input will be a single integer, N, which indicates the number of input sets to process. Each of the input set will contain one line giving the race of each player, separated by the letter ‘v’. Boudreaux’s race is indicated first, followed by a ‘v’, then Thibodeaux’s race. Races are given as a one letter code: ‘T’ for Terran, ‘P’ for Protoss, and ‘Z’ for Zerg.

Output:
For each input set, output a single line containing either the word “Boudreaux”, “Thibodeaux”, or “Tie” to predict who will win that game, as described in the information above. The output for each input set should be on a separate line.

Sample Input:

```
2
PvT
PvZ
```

Sample Output:

```
Boudreaux
Tie
```

Problem 9 - Gator Skin Boots

Taking out alligators on their paths to work using the “conventional” method from problem 3 hasn’t worked so well for Boudreaux and Thibodeaux. As such, they have decided to take things up a notch from guns to dynamite. They have located a number of gators that hang around the area they care about and have tracked their motions so they know their habits. All that is left is planning the details of their attack. They are still on a budget, so they want to do the deed with as few sticks of dynamite as possible. To figure out how many that is, they need a program.

The alligators move (or don’t) one distance unit each time period. Their positions are always integer values. A stick of dynamite can be set off (or not) each time period and they have to be located at an integer coordinate pair (x,y). When set off, a stick will kill any alligators in the 3x3 square area around the location at that time.

Input: The input will begin with a number telling you how many data sets there are. Each data set begins with a line that has a single number, N (0<N<11), telling you how many alligators there are. After that will be N lines that start with two integers followed by strings with the characters ‘u’, ‘d’, ‘l’, ‘r’, and ‘-’. The integers are the starting location of the alligator. The letters stand for movement up, down, left, and right respectively. A ‘-’ means the alligator doesn’t move.

To see how this works, consider the following input line.

4 5 lr-

This alligator starts at (4,5). Between the 1st and 2nd time period it moves one unit left to (3,5) and is at that location for the second time period. For period 3 it is back at (4,5), having moved to the right. It doesn’t go anywhere for period 4 so it stays at (4,5). Before period 5 it moves back to the left and the cycle repeats. Note that the initial conditions must be considered for the first possible detonation. Also, the length of the direction string can be different for each alligator.

Output: For each input your will output a single line with the number of sticks of dynamite they have to buy to be able to take out all the alligators for that input in 100 time units.

Sample Input:

```
3
2
2 0 llrr
-2 0 rrll
```

```
2
2 0 rrll
-2 0 llrr
3
2 0 llrr
-2 0 rrll
0 5 lurd--
```

Sample Output:

```
1
2
2
```