



## Canadian Mathematics Competition

An activity of The Centre for Education  
in Mathematics and Computing,  
University of Waterloo, Waterloo, Ontario

# *Canadian Computing Competition*

for the  SYBASE® Awards

**Tuesday, February 29, 2000**

## Problem S1J3

### Slot Machines

Martha takes a jar of quarters to the casino with the intention of becoming rich. She plays three machines in turn. Unknown to her, the machines are entirely predictable. Each play costs one quarter. The first machine pays 30 quarters every 35th time it is played; the second machine pays 60 quarters every 100th time it is played; the third pays 9 quarters every 10th time it is played.

Your program should take as input the number of quarters in Martha's jar (there will be at least one and fewer than 1000), and the number of times each machine has been played since it last paid.

Your program should output the number of times Martha plays until she goes broke.

Input/output is not from/to files for this question. Keyboard input and screen output is expected.

#### **Sample Session.** *User input is in italics*

How many quarters does Martha have in the jar?

*48*

How many times has the first machine been played since paying out?

*3*

How many times has the second machine been played since paying out?

*10*

How many times has the third machine been played since paying out?

*4*

Martha plays 66 times before going broke.

## Problem S2J4

### Babbling Brooks

A series of streams run down the side of a mountain. The mountainside is very rocky so the streams split and rejoin many times. At the foot of the mountain, several streams emerge as rivers. Your job is to compute how much water flows in each river.

At any given elevation there are  $n$  streams, labelled 1 to  $n$  from left-to-right. As we proceed down the mountainside, one of the streams may split into a left fork and a right fork, increasing the total number of streams by 1, or two streams may rejoin, reducing the total number of streams by 1. After a split or a rejoining occurs, the streams are renumbered consecutively from left-to-right. There is always at least one stream and there are never more than 100 streams.

The first line of input contains  $n$ , the initial number of streams at some high altitude. The next  $n$  lines give the flow in each of the streams from left-to-right. Proceeding down the mountainside, several split or rejoin locations are encountered. For each **split** location, there will be three lines of input;

- a line containing 99 (to indicate a split)
- a line containing the number of the stream that is split
- a line containing a number between 0 and 100, the percentage of flow from the split stream that flows to the left fork. (The rest flows to the right fork).

For each **join** location, there will be two lines of input;

- a line containing 88 (to indicate a join)
- a line containing the number of the stream that is rejoined with the stream to its right

The flow from both joined streams is combined. After the last split or join location will be:

- a single line containing 77 (to indicate end of input)

Your job is to determine how many streams emerge at the foot of the mountain and what the flow is in each. Your output is a sequence of real numbers, rounded to the nearest integer, giving the flow in rivers 1 through  $n$ .

**Sample Input** (Input file : brooks.in)

```
3
10
20
30
99
1
50
88
3
88
2
77
```

## **Output for Sample Input** (Output file : brooks.out)

5 55

## Problem S3J5

### Surfin'

Every Web Page is identified by a string of characters known as a URL (Uniform Resource Locator). Web Pages are formatted using HTML (Hypertext Markup Language). HTML has many codes, collectively known as markup, that allow the author to specify the format of the pages as well as to specify links to other pages. For this problem, we are concerned only with the markup that identifies links to other pages within a given page.

A link within the page is denoted `<A HREF="URL">` where *URL* is the URL of some other page. A user viewing a page containing a link may click on the link to view the other page.

You are to write a program that reads a number of web pages and their associated URLs. For each link in each page, you must print the URL of the page containing the link, and the URL of the page referred to by the link.

Following the last page, you are then given several pairs of URLs. For each pair you are to assume that you are viewing the page identified by the first URL, and determine whether it is possible to click a sequence of links so as to view the page identified by the second URL. If so, you should print "Can surf from *here* to *there*." where *here* and *there* are the two URLs. If not you should print "Can't surf from *here* to *there*."

The first line of input contains an integer  $n \leq 100$ , the number of Web Pages. For each Web Page, there will be a line containing its URL, followed by several lines containing the page. The URL will consist of up to 80 non-blank printable characters and will not contain any quotation marks. The first line of the page will be `<HTML>` and the last line will be `</HTML>`. Each page will contain up to 100 links in the format described above. Each link will be contained within a single line of input. URLs in the link will be those of pages given in the input. The markup keywords A, HREF, and HTML will appear only in upper case.

Following the  $n$  pages will be several pairs of lines giving URLs required by the problem as specified above. The last line of input will be "The End". For each pair, print the appropriate message given above.

#### Sample Input ( Input file : surf.in )

```
3
http://ccc.uwaterloo.ca
<HTML> <TITLE>This is the CCC page</TITLE>
Hello there boys
and girls.  <B>Let's</B> try <A HREF="http://abc.def/ghi"> a
little
problem </A>
</HTML>
http://abc.def/ghi
<HTML> Now is the <TITLE>time</TITLE> for all good people to program.
<A HREF="http://www.www.www.com">hello</A><A HREF="http://xxx">bye</A>
```

```
</HTML>
http://www.www.www.com
<HTML>
<TITLE>Weird and Wonderful World</TITLE>
</HTML>
http://ccc.uwaterloo.ca
http://www.www.www.com
http://www.www.www.com
http://ccc.uwaterloo.ca
The End
```

### **Output for Sample Input ( Output file : surf.out )**

Link from http://ccc.uwaterloo.ca to http://abc.def/ghi  
Link from http://abc.def/ghi to http://www.www.www.com  
Link from http://abc.def/ghi to http://xxx

Can surf from http://ccc.uwaterloo.ca to http://www.www.www.com.  
Can't surf from <http://www.www.www.com>

## Problem S4

### Golf

Roberta the Robot plays a perfect game of golf. When she hits the golf ball it always goes directly towards the hole on the green, and she always hits exactly the distance that is specified for the club. Each such action is known as a stroke, and the object of golf is to hit the ball from the tee to the hole in the fewest number of strokes. Roberta needs a program to select the best combination of clubs to reach the hole in the fewest strokes. She also needs to decide if the task is impossible, in which case she graciously acknowledges the loss. Roberta can carry up to 32 clubs, and the total distance from the tee to the hole does not exceed 5280 metres.

#### The Input

The first line of input gives the distance from the tee to the hole, an integral number of metres between 1 and 5280. The next line states the number of clubs, between 1 and 32. For each club, a line follows giving the distance, in metres, that the club will hit the ball, an integer between 1 and 100. No two clubs have the same distance.

#### The Output

If Roberta can get the ball from the tee to the hole, without passing the hole, print "Roberta wins in  $n$  strokes." where  $n$  is minimized. If Roberta cannot get the ball from the tee to the hole, print "Roberta acknowledges defeat."

#### Sample Input ( Input file : golf.in )

```
100
3
33
66
1
```

#### Output for Sample Input ( Output file : golf.out )

```
Roberta wins in 3 strokes.
```

## Problem S5

### Sheep and Coyotes

A square 1000 by 1000 field contains several sheep. A coyote enters the field at some point in the south boundary and proceeds to eat the sheep closest to the point of entry, picking arbitrarily if more than one sheep is equally close. The coyote, being sated, then leaves the field.

Your job is to determine which sheep may be eaten by the coyote.

Assume that the southwest corner of the field is located at (0.00, 0.00), the northwest corner at (0.00, 1000.00), the northeast corner at (1000.00, 1000.00) and the southeast corner at (1000.00, 0.00). The first line of input gives the number of sheep, between 1 and 100. For each sheep a pair of lines follows, giving its coordinates within the field (between 0.00 and 1000.00).

For each sheep that might be eaten print a line "The sheep at (x, y) might be eaten." where  $x$  and  $y$  give the location of the sheep to two decimal places. The sheep can be listed in any order in the output.

#### Sample Input ( Input file : sheep.in )

```
6
100.00
100.00
200.00
150.00
140.00
200.00
100.00
300.00
300.00
300.00
300.00
100.00
```

#### Output for Sample Input ( Output file : sheep.out )

```
The sheep at (100.00, 100.00) might be eaten.
The sheep at (300.00, 100.00) might be eaten.
```