

# ACM Programming Contest

## Northeast North America Region

Western Preliminary

@

SUNY Oswego  
October 14, 2006

## Contest Questions

### Problem 1: IM Poster Imposter

As part of the class pedagogy, your professor requires that all students participate in an on-line discussion group. The professor suspects that one student isn't contributing, rather that a fellow student is signing on and posting various messages as both users.

The professor assumes that the two sets of messages posted by the same user will have more words in common than any other pair of sets of messages posted to the group. Your task is to write code to determine which two users have the "highest degree of overlap" in word usage. The specific measure of "degree of overlap" to be used is the ratio of distinct common words to total distinct words in the two groups.

Case is ignored; punctuation, and any token not one of a-z or 0-9, is removed and not treated as a word boundary. So, for example, the word The\_Decider\$ is equivalent to the word thedecider.

#### Input

Input will be an unknown number of lines, each of the form:

```
user-name: list of words in post separated by spaces
```

Each post will be contained on a single line. There are an unknown number of distinct users.

**As an example, consider the exchange:**

```
Van:    cool chat topic.
Gwen:  sorry to be a PITA, but what's today's topic?
Bluto:  queues.
Otter:  'Cuse? The orangemen? LOL
Bluto:  what's a PITA?
Van:    you can get a veggie pita for $5. LOL
Gwen:  can we get back to queues?
```

The 4 users give us the following pairs:

```
(Van, Gwen), (Van, Bluto), (Van, Otter),
(Gwen, Bluto), (Gwen, Otter), (Otter, Bluto)
```

Van uses 12 distinct words:

```
{5 a can chat cool for get lol pita topic veggie you}
```

gwen uses 16 words, with only 15 distinct:

```
{a back be but can get pita queues sorry the to todays topic we whats}
```

Otter uses 4 words: {cuse lol orangemen the}

Bluto uses 4 words: {a pita queues whats}

The pair (Van,Gwen) uses 22 distinct words with 5 words of overlap for a ratio of  $5/22 = 0.23$ . The pair (Van,Bluto) uses 14 distinct words with 2 words of overlap for a ratio of  $2/14 = 0.14$ . The pair (Van,Otter) uses 15 distinct words with 1 word of overlap for a ratio of  $1/15 = 0.07$ . The pair (Gwen,Bluto) uses 15 distinct words with 4 words of overlap for a ratio of  $4/15 = 0.27$ . The pair (Gwen,Otter) uses 18 distinct words with 1 word overlap for a ratio of  $1/18 = 0.05$ . The pair (Otter,Bluto) uses 8 distinct words with 0 overlap.

So, with a precision way beyond the reasonableness of our measure we can state that Bluto and Gwen are one and the same student.

#### Output

The names of the users who are deemed to be the same are displayed.

## Problem 2: Stakes on a Plane

A building site is laid out on a planar rectangular grid of M units by N units. K customers have ordered homes, each with a specified area for its footprint. You are to place stakes on the grid to show the outline of each home, or specify that no arrangement is possible to accommodate all the required homes.

Stakes can be placed only at multiples of integral unit points. Each house has a convex rectangular shape (i.e., no L-shaped or U-shaped or other irregularly formed homes; just rectangles, including squares.)

**Input** consists of K+1 lines:

- line one contains M and N separated by a space
- each of the next K lines has a single integer specifying the square units needed for each home

Input example:

```
5 12
5
10
9
```

**Output** should be an ASCII image of the grid with the stakes suitably

placed. Each stake is represented by a number corresponding to the order in which that house's area was input. Grid points with no stakes should be represented with a lower case letter 'o'. There should be no spaces between the grid points

For example if a 5 X 12 grid were specified, you might have a mental image similar to that below, where the 'o's denote where stakes may be placed. Note that 5, the first number, denotes the number of units going down the page, while 12 is the number of units across. Note also, that the stake locations form a 6 x 13 grid since measurement starts at zero.

```
oooooooooooo
oooooooooooo
oooooooooooo
oooooooooooo
oooooooooooo
oooooooooooo
```

If you had to place three homes with areas 5, 10, and 9 on the lot, as shown above as our example input, one solution would be:

```
11111122222o
1111112oooo2o
oooooo22222o
3333333333ooo
3333333333ooo
oooooooooooo
```

Notice that the 1's outline an area of 5 units, the 2's outline the home with 10 unit area, and the 3's the area for the home of 9 square units, since that is the order in which the data appeared.

## Problem 3: Lengths and Limits

### The Setup

The *mystery sequence* of numbers for positive integer x is defined by the following process:

```
to compute mysseq(X) do
  if ( X = 1) then
    record X
    stop
  else if ( X is even ) then
    record X
    mysseq(X/2)
  else
    record X
    mysseq((3*X)+1)
  end
end
```

Thus, for example:

- mysseq(1) = < 1 >
- mysseq(2) = < 2 1 >
- mysseq(3) = < 3 10 5 16 8 4 2 1 >
- mysseq(4) = < 4 2 1 >
- mysseq(5) = < 5 16 8 4 2 1 >
- mysseq(6) = < 6 3 10 5 16 8 4 2 1 >
- mysseq(7) = < 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 >
- mysseq(8) = < 8 4 2 1 >
- mysseq(9) = < 9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 >

- `mysseq(10) = < 10 5 16 8 4 2 1 >`
- `mysseq(11) = < 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 >`
- `mysseq(12) = < 12 6 3 10 5 16 8 4 2 1 >`
- `mysseq(13) = < 13 40 20 10 5 16 8 4 2 1 >`
- `mysseq(14) = < 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 >`
- `mysseq(15) = < 15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1 >`

For any given positive integer  $x$  the *length* and *limit* of the mystery sequence corresponding to  $x$  are defined as follows:

- `length(mysseq(x))` = the number of elements in the sequence
- `limit(mysseq(x))` = the highest number in the sequence

## The Problem

Write a program to compute and display, labeled, the following two numbers:

1. The maximum length of the sequences corresponding to the values from 1 to 5000.
2. The maximum limit of the sequences corresponding to the values from 1 to 5000.

(Note that for the sequences corresponding to the values from 1 to 15 the maximum length is 20 (sequence 9) and the maximum limit is 160 (sequence 15).)

## Problem 4: Traveling Insect Problem

### The Setup

An insect is to fly in a sequence of shortest distance straight lines from the origin to a number of points in 3-space and back to the origin. Your job is to find a sequence of the points which produces a shortest distance for the entire trip.

### The Problem

Write a program consistent with the following input/output specification.

#### • Input

A positive integer followed by that number of points in three space. Assume that each point takes the form of a left parenthesis followed by three integer values between -100 and 100, each separated by a space, followed by a right parenthesis. For example, the following are possible data sets:

- 1 (1 1 1)
- 2 (1 2 3)(3 2 1)
- 3 (62 19 -92)(5 -93 32)(-94 -97 -54)
- 4 (-58 -12 66)(-90 57 54)(94 -41 -69)(-50 38 26)
- 5 (1 2 3)(8 80 3)(1 1 1)(11 5 3)(-3 -4 -5)
- 6 (-48 -61 24) (63 6 36) (-62 27 86) (62 19 -92) (5 -93 32) (-94 -97 -54)

#### • Output

The distance of a shortest route which starts at the origin, goes to each of the given points (in whichever order produces a shortest route), and returns to the origin.

#### • Example

If the input is

```
5 (1 2 3)(8 80 3)(1 1 1)(11 5 3)(-3 -4 -5)
```

then the answer would be 181.6 since the route (0 0 0) --> (1 1 1) --> (1 2 3) --> (11 5 3) --> (8 80 3) --> (-3 -4 -5) --> (0 0 0) is of distance 181.6 and there is no shorter route by which the straight flying insect can visit each point, beginning and ending at the origin.

## Problem 5: Palindromic Sentences

You've just been hired as Secur's new computer security guru. Your boss thinks that the new security breaches will be in the form of text files that contain palindromic sentences. A sentence ends in any of ".", "!", or "?".

Your job is to find and report on all sentences in a file:

If the sentence is a palindrome, output "YES" followed by the sentence, otherwise output "NO" followed by the sentence. Notice that a sentence may span more than one line.

Input:

```
Go hang a salami, I'm a lasagna hog! Eve damned Eden (mad Eve!).
```

```
Bosnia gasps again -- sob! Er was I saw Elba. Stressed was I, sad, alas, to order a redroot salad as I saw desserts. He was, was he? I did, did I
```

Output:

```
YES Go hang a salami, I'm a lasagna hog!
```

```
YES Eve damned Eden (mad Eve!).
```

```
YES Bosnia gasps again -- sob!  
NO Er was I saw Elba.  
YES Stressed was I, sad, alas, to order a redroot salad as I saw desserts.  
YES He was, was he?  
YES I did, did I?
```

## Problem 6: Hotter Colder

The children's game *Hotter Colder* is played as follows. Player *A* leaves the room while player *B* hides an object somewhere within. Player *A* re-enters at position (0, 0) and then visits various other locations about the room. When player *A* visits a new position, player *B* announces "Hotter" if this position is closer to the object, than the previous position, "Colder" if it is farther, and "Same" if it is the same distance.

### Input

Input consists of up to 50 lines, each containing an (x, y)-coordinate pair followed by "Hotter", "Colder", or "Same". Each pair represents a position within the room, which may be assumed to be a square with opposite corners at (0, 0) and (10, 10).

### Output

For each line of input, print a line giving the total area of the region in which the object may have been placed, to two decimal places. If there is no such region, output "0.00".

### Sample Input

```
10.0 10.0 Colder  
10.0 0.0 Hotter  
0.0 0.0 Colder  
10.0 10.0 Hotter
```

### Sample Output

```
50.00  
37.50  
12.50  
0.00
```