# ECOO 2014

# Programming Contest Questions

Final Competition (Round 3)

May 10, 2014

Sheridan | Get Creative

Questions made possible in part through the support of the Sheridan College Faculty of Applied Science and Technology.

# Problem 1: Substitute Teacher

Your math teacher has created an encrypted message using a substitution cipher. The first person to crack the encryption and find the original message gets to leave class early.

```
pIZjugwgZ6HkZx6kZjiZg77GtGgHjUkZ76tjiwZ6ZIgvG9wGvgZI tuZ6IZY,LLYLKln
```

She explained her encryption scheme as follows. First she converts every character in the original message to an integer. The capital letters **A** to **Z** are encoded as the integers 1 to 26, the lowercase letters **a** to **z** are encoded as the integers 27 to 52, the digit characters **0** to **9** are encoded as the integers 53 to 62 and the space, period, comma and question mark are 63, 64, 65 and 66 respectively.

Then she uses the formula $\boldsymbol{c} = \mathbf{rem}(\boldsymbol{mk_e}, \mathbf{67})$ to substitute each character in the original message for a new one. In this formula, $\boldsymbol{m}$ is the character number from the original message, $\boldsymbol{k_e}$ is a secret encryption key ($\mathbf{2} \le \boldsymbol{k_e} \le \mathbf{66}$) and $\boldsymbol{c}$ is the number for the character in the encrypted message and the $\mathbf{rem}$ function gives the remainder after dividing the first argument by the second. The formula maps every integer from 1 to 66 to a different integer in the same range with no repeats. A similar formula will also work for decrypting: $\boldsymbol{m} = \mathbf{rem}(\boldsymbol{ck_e}, \mathbf{67})$ where $\boldsymbol{k_d}$ is a decryption key ($\mathbf{2} \le \boldsymbol{k_d} \le \mathbf{66}$).

If $\boldsymbol{k_e} = \mathbf{18}$ then an '**a**' character (27) in the original message would become a '**Q**' character (17) in the encrypted message, because $\boldsymbol{rem}(\mathbf{27} \times \mathbf{18}, \mathbf{67}) = \mathbf{17}$. For this example the decryption key is $\boldsymbol{k_d} = \mathbf{41}$.

You just found a scrap of paper on the floor where the teacher encrypted her name. So you now know that '**B6Hg**' is the encrypted word '**Jane**'. Can you use this information to decrypt the longer message?

DATA11.txt (DATA12.txt for the second try) will contain 10 test cases. Each test case consists of 3 lines. The first two lines are a short sample message (unencrypted version, then encrypted version) and the third is the message you have to decrypt using the same method that works for the sample. All three strings will start and end with an asterisk character which is not part of the message. The maximum string length is 200 characters, including asterisks. Your job is to use the sample to decrypt the longer message and then print it out between asterisks. Note that there are only two test cases below, but the real data sets will contain 10 test cases.

### *Sample input*

(a single space character)

```
*Jane*
*B6Hg*
*pIZjugwgZ6HkZx6kZjiZg77GtGgHjUkZ76tjiwZ6ZIgvG9wGvgZI tuZ6IZY,LLYLKln*
*135*
*29B*
*,40Va0aT.H0az8a0Y08Fl0Y0kpuk1g08ll0aq0fdddd*
```

### *Sample Output*

```
*Is there any way to efficiently factor a semiprime such as 54775723?*
*Is it true that 2 and 2 ALWAYS add to 4????*
```

# Problem 2: Mothers and Daughters

In the land of Matriarchia, lineages are traced only through the mothers. Women derive their status from their female family relations and the only relationships that matter are mother, daughter, sister and cousin. Every woman has one mother. A woman is sister to another woman if they have the same mother. A woman is cousin to another woman if they are not sisters and their mothers have the same mother.

DATA21.txt (DATA22.txt for the second try) will contain an integer $N$, followed by $N$ lines ($1 \leq N \leq 1000$). Each line will contain two names separated by a space. The first name is a mother and the second name is one of her daughters. Names can appear multiple times, but no name will appear more than once in the daughter position. There may be as many as 1000 distinct names in this list. Because the list is not infinite, not every woman will have her mother listed.

The list of mother-daughter pairs will be followed by 10 lines, each containing a single name from the preceding list. Your job is to compute the number of cousins and sisters each of these 10 women has and output the result using the exact format shown below. There will always be enough information in the mother-daughter pair list to compute this information.

***Sample Input***

```
19
Jenna Breanna
Jacklyn Sophie
Tanisha Jenna
Jacklyn Emily
Jacklyn Lindsey
Jenna Kailee
Justina Malia
Kailee Hillary
Kailee Myranda
Jenna Shayna
Justina Tatiana
Jenna Raquel
Kailee Mary
Justina Mayra
Tanisha Yasmeen
Jacklyn Darian
Justina Destiny
Jenna Jacklyn
Tanisha Justina
Breanna
Darian
Destiny
Malia
Mary
Mayra
Raquel
Tatiana
Sophie
Myranda
```

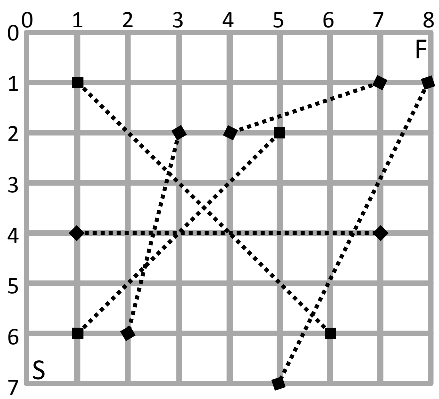
***Sample Output***

```
Cousins: 4, Sisters: 4
Cousins: 3, Sisters: 3
Cousins: 5, Sisters: 3
Cousins: 5, Sisters: 3
Cousins: 4, Sisters: 2
Cousins: 5, Sisters: 3
Cousins: 4, Sisters: 4
Cousins: 5, Sisters: 3
Cousins: 3, Sisters: 3
Cousins: 4, Sisters: 2
```

# Problem 3: Future City

The cities of the future will be laid out in perfect grids. Nobody will drive or take the bus. Transit problems will have been solved once and for all by placing teleportation devices (TDs) at intersections. Pedestrians entering an intersection containing a TD will have the option of continuing through the intersection as normal or using an app on their phone to trigger the TD and instantly teleport themselves to a new intersection. The TDs will be linked in pairs as shown below. Either TD in the pair can be used to instantly teleport a pedestrian to the other TD.

Intersections are identified using a pair of integers representing the horizontal street number (numbered North to South starting at $0$) and then the vertical street number (numbered West to East starting at $0$).




Suppose a future pedestrian wants to go from the point marked $S$ $(7,0)$ to the point marked $F$ $(0,8)$ on the map shown at left. She could use the TD at $(6,1)$ to instantly teleport to $(2,5)$. If she does that, her total distance will be 7 blocks: she walks 2 blocks from $S$ to $(6,1)$ then 5 from $(2,5)$ to $F$. If she takes the TD at $(7,5)$, she can shorten her route to 6 blocks. But there is also a combination of 2 TDs that can get her to $F$ in 5 blocks. Can you find it?

DATA31.txt (DATA32.txt for the second try) will contain 10 test cases. The first line of each test case consists of a pair of numbers H and V which give the number of horizontal and vertical streets in a future city ($1 \le H, V \le 1000$). This is followed by a line with two integers $S_h$ and $S_v$ representing the start location of a pedestrian and another line with two integers $F_h$ and $F_v$ representing their finishing location ($0 \le S_h, S_v, F_h, F_v \le 999$). This is followed by a line with a single integer N which gives the number of pairs of TDs ($0 \le N \le H \cdot V/4$) and this is followed by N lines, each containing four integers $T_h$, $T_v$, $U_h$ and $U_v$ representing the locations of the two TDs in each pair. Each intersection can hold only one TD. Your job is to find the shortest path between the pedestrian's start and finish locations and report the number of city blocks the pedestrian will have to walk. The path can make use of any number of teleportation devices. Note that the data set below contains only 3 test cases, but the real data set will contain 10.

### *Sample Input*

```
8 9
7 0
0 8
6
1 1 6 6
4 1 4 7
6 1 2 5
2 3 6 2
1 8 7 5
2 4 1 7
5 4
4 0
0 3
2
2 1 0 3
4 1 2 2
1000 1000
5 2
10 970
14
0 3 4 5
1 1 13 13
1 2 2 3
7 7 9 9
1 70 70 1
9 1 1 12
2 2 3 3
3 7 2 900
3 900 5 95
3 90 5 100
4 5 5 4
6 7 8 9
9 9 8 8
9 9 5 3
```

### *Sample Output*

```
5
2
83
```

# Problem 4: Baby Talk

When babies babble they say things like GAGAGOOGOO or BABABABA. For the purposes of this question we will define a baby talk word to be any non-empty string of letters which can be divided into two equal-length portions in such a way that the first portion is identical to the second.

Based on that definition, the following strings are words in baby talk: GAGA, GOOGOO, BABA, GUBBAGUBBA, DOGGIEDOGGIE, FDSFDS, IWANTMOREMILKIWANTMOREMILK and XX.

The following strings are not words in baby talk: BABAB, GAGOO, BA, DOGGIE and X.

Based on this definition, the string GAGAGOOGOO is actually two baby talk words concatenated together. The string BABABABA could be a single baby talk word or it could be two baby talk words, depending on how you choose to divide it.

DATA41.txt (DATA42.txt for the second try) will contain 10 test cases. Each test case consists of a single line of text containing a string of capital letter characters of length N ($1 \le N \le 2000$). Your job is to find the longest substring consisting only of consecutive baby talk words, as defined above, and report the length of that substring on a single line. In the test cases below, the longest baby talk string in each input string is underlined.

### *Sample Input*

```
GOOGOOGAGA
BABABABA
PTHHPTHHBAGOOGOOGAGABOOOOO
XYBABABABAXYX
GOOGOOGAGAGAGAGOOOGAGAGOOO
BABABABABA
CUTEBABYTALKSTRING
CCEEKCKCBFBFEEBABAFKFKAADDDAAKKBBEEIIHHDDGGIIEEIIDDHHGGBBHHKKJJHHCCAAJJDDDAAKKBB
IEIEFFFFAAJJBBJJJ
BBHHBBBBFFBGGEEEEEEHHAAIIACC
```

### *Sample Output*

```
10
8
10
8
26
8
0
48
16
14
```

| **Question Development Team** | **President of ECOO-CS** |
|---|---|
| Sam Scott (Sheridan College) | David Stermole |
| Kevin Forest (Sheridan College) | |
| Greg Reid (St. Francis Xavier Secondary School, Mississauga) | **Communications** |
| Dino Baron (University of Waterloo) | John Ketelaars |
| Nathan Schucher (University of King's College) | |