

# American Computer Science League

\* 1996–97 All-Star Contest \* East HS, Salt Lake City, UT \*

Dear Student:

The following pages contain the All-Star Contest programming problems for all divisions. You may do any of the problems you'd like; however, your team will only be entitled to receive points for programs solved from among those designated for its division. The designations are as follows:

Junior Division:

- |                 |        |
|-----------------|--------|
| 1. TimeCards    | 5 pts  |
| 2. Botchagaloop | 5 pts  |
| 3. Digit Count  | 10 pts |
| 4. You Are El   | 10 pts |

Intermediate Division (3-member teams):

- |                           |        |
|---------------------------|--------|
| 3. Digit Count            | 10 pts |
| 4. You Are El             | 10 pts |
| 6. The Biggest, Revisited | 10 pts |
| 7. Squarepoint            | 10 pts |

Intermediate Division (5-member teams):

- |                               |        |
|-------------------------------|--------|
| 3. Digit Count                | 10 pts |
| 4. You Are El                 | 20 pts |
| 5. A Balancing Act, Revisited | 10 pts |
| 6. The Biggest, Revisited     | 10 pts |
| 7. Squarepoint                | 10 pts |

Senior Division (3-member teams):

- |                           |        |
|---------------------------|--------|
| 4. You Are El             | 10 pts |
| 6. The Biggest, Revisited | 10 pts |
| 7. Squarepoint            | 10 pts |
| 8. Alphaddition           | 10 pts |

Senior Division (5-member teams):

- |                               |        |
|-------------------------------|--------|
| 4. You Are El                 | 20 pts |
| 5. A Balancing Act, Revisited | 10 pts |
| 6. The Biggest, Revisited     | 10 pts |
| 7. Squarepoint                | 10 pts |
| 8. Alphaddition               | 10 pts |

In all programs each output is worth one point, with one exception: for the 5-member Intermediate and 5-member Senior teams, each output in Program 4. You Are El, is worth 2 points. By and large, the sample data shows only part of the input. That is, if the test data will consist of 10 sets of data, for instance, the sample data may show only 2 sets. Be careful! Also, if you don't print the outputs in order, be sure to label them.

Good luck, and have fun!

# American Computer Science League

\* 1996–97 All-Star Contest \* East HS, Salt Lake City, UT \*

## Junior Division Programming Problems

- |                 |        |
|-----------------|--------|
| 1. TimeCards    | 5 pts  |
| 2. Botchagaloop | 5 pts  |
| 3. Digit Count  | 10 pts |
| 4. You Are El   | 10 pts |

In all programs each output is worth one point. By and large, the sample data shows only part of the input. That is, if the test data will consist of 10 sets of data, for instance, the sample data may show only 2 sets. Be careful, since **you may run your program only once on our data!** Also, if you don't print the outputs in order, be sure to label them.

# American Computer Science League

\* 1996–97 All-Star Contest \* East HS, Salt Lake City, UT \*

|                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------|
| <p style="text-align: center;"><b>Intermediate Division (3-member teams)</b><br/><b>Programming Problems</b></p> |
|------------------------------------------------------------------------------------------------------------------|

|                                  |               |
|----------------------------------|---------------|
| <b>3. Digit Count</b>            | <b>10 pts</b> |
| <b>4. You Are El</b>             | <b>10 pts</b> |
| <b>6. The Biggest, Revisited</b> | <b>10 pts</b> |
| <b>7. Squarepoint</b>            | <b>10 pts</b> |

In all programs each output is worth one point. By and large, the sample data shows only part of the input. That is, if the test data will consist of 10 sets of data, for instance, the sample data may show only 2 sets. Be careful, since **you may run your program only once on our data!** Also, if you don't print the outputs in order, be sure to label them.

# American Computer Science League

\* 1996–97 All-Star Contest \* East HS, Salt Lake City, UT \*

|                                                                                                            |
|------------------------------------------------------------------------------------------------------------|
| <p style="text-align: center;"><b>Senior Division (3-member teams)</b><br/><b>Programming Problems</b></p> |
|------------------------------------------------------------------------------------------------------------|

- |                                  |               |
|----------------------------------|---------------|
| <b>4. You Are El</b>             | <b>10 pts</b> |
| <b>6. The Biggest, Revisited</b> | <b>10 pts</b> |
| <b>7. Squarepoint</b>            | <b>10 pts</b> |
| <b>8. Alphaddition</b>           | <b>10 pts</b> |

In all programs each output is worth one point. By and large, the sample data shows only part of the input. That is, if the test data will consist of 10 sets of data, for instance, the sample data may show only 2 sets. Be careful, since **you may run your program only once on our data!** Also, if you don't print the outputs in order, be sure to label them.

# American Computer Science League

\* 1996–97 All-Star Contest \* East HS, Salt Lake City, UT \*

|                                                                          |
|--------------------------------------------------------------------------|
| <h2>Intermediate Division (5-member teams)<br/>Programming Problems</h2> |
|--------------------------------------------------------------------------|

|                               |        |
|-------------------------------|--------|
| 3. Digit Count                | 10 pts |
| 4. You Are El                 | 20 pts |
| 5. A Balancing Act, Revisited | 10 pts |
| 6. The Biggest, Revisited     | 10 pts |
| 7. Squarepoint                | 10 pts |

In all programs each output is worth one point, with one exception: each output in *Program 4. You Are El* is worth 2 points. By and large, the sample data shows only part of the input. That is, if the test data will consist of 10 sets of data, for instance, the sample data may show only 2 sets. Be careful, since **you may run your program only once on our data!** Also, if you don't print the outputs in order, be sure to label them.

# American Computer Science League

\* 1996–97 All-Star Contest \* East HS, Salt Lake City, UT \*

|                                                                    |
|--------------------------------------------------------------------|
| <h2>Senior Division (5-member teams)<br/>Programming Problems</h2> |
|--------------------------------------------------------------------|

- |                               |        |
|-------------------------------|--------|
| 4. You Are El                 | 20 pts |
| 5. A Balancing Act, Revisited | 10 pts |
| 6. The Biggest, Revisited     | 10 pts |
| 7. Squarepoint                | 10 pts |
| 8. Alphaddition               | 10 pts |

In all programs each output is worth one point, with one exception: each output in *Program 4. You Are El* is worth 2 points. By and large, the sample data shows only part of the input. That is, if the test data will consist of 10 sets of data, for instance, the sample data may show only 2 sets. Be careful, since **you may run your program only once on our data!** Also, if you don't print the outputs in order, be sure to label them.

## American Computer Science League

---

### \* 1996–97 All-Star Contest \* 1. TimeCards – 5 Points \*

**Problem:** Jenny just started work as a programmer for Justine’s Java Workshop. She is paid \$10 an hour, with a few exceptions. She earns an extra \$1.50 an hour for any part of a day where she works more than 8 hours, and an extra \$2.50 an hour for hours beyond 40 in any one week. Also, she earns a 125% bonus for working on Saturday, and a 50% bonus for working on Sunday. The bonuses for Saturday and Sunday are computed based on the hours worked those days; they are not used to calculate any bonus for working more than 40 hours in a week.

You’ll be given the number of hours Jenny worked each day in a week (Sunday, Monday, . . . , Saturday), and you need to compute her salary for the week. The input will be positive integers, less than or equal to 24. The output must be formatted with a dollar sign and rounded up to the nearest penny. For example, “\$2” and “\$2.136666” are wrong answers; the correct versions are “\$2.00” and “\$2.14”, respectively. There may not be any embedded spaces in your answers.

There will be 5 sets of data.

**Sample Input:**

Line #1: 0, 8, 8, 8, 10, 6, 0

Line #2: 4, 0, 0, 0, 0, 6, 0

...

**Sample Output:**

Output #1: \$403.00

Output #2: \$120.00

...

# American Computer Science League

---

\* 1996–97 All-Star Contest \* 1. TimeCards – 5 Points \*

**Test Input:**

Line #1: 0, 8, 8, 8, 8, 8, 0

Line #2: 0, 10, 10, 10, 10, 10, 0

Line #3: 0, 0, 8, 8, 8, 8, 8

Line #4: 0, 0, 0, 10, 10, 10, 10

Line #5: 10, 10, 10, 9, 9, 9, 9

**Test Output:**

Output #1: \$400.00

Output #2: \$540.00

Output #3: \$500.00

Output #4: \$540.75

Output #5: \$905.88



## American Computer Science League

---

### \* 1996–97 All-Star Contest \* 2. Botchagaloop – 5 Points \*

**Problem:** The *botchagaloop* value of a number  $x$  is found as follows. First, convert  $x$  to base 8. Call this  $p$ . Next, sort the digits of  $p$  in increasing order. Call this  $q$ . Subtract  $q$  from  $p$  (in base 8, of course). Repeat the “sort-subtract” sequence 4 more times, or until the digits in the result are in sorted order (whichever come first). Finally, convert the number back to base 10.

For example, 3418 has a botchagaloop value of 1008. It is computed as follows:  $3418 = 6532_8$ ;  $6532 - 2356 = 4154$ ;  $4154 - 1445 = 2507$ ;  $2507 - 257 = 2230$ ;  $2230 - 223 = 2005$ ;  $2005 - 25 = 1760$ ; and finally,  $1760_8 = 1008$ . Note that there is at least one subtraction and at most 5 subtractions.

There will be 5 inputs. Each is a positive integer less than 1,000,000. Print the botchagaloop value of each input.

**Sample Input:**

Line #1: 3418

Line #2: 123

...

**Sample Output:**

Output #1: 1008

Output #2: 28

...

# American Computer Science League

---

\* 1996–97 All-Star Contest \* 2. Botchagaloop – 5 Points \*

**Test Input:**

Line #1: 251

Line #2: 5049

Line #3: 14711

Line #4: 23328

Line #5: 199697

**Test Output:**

Output #1: 28

Output #2: 301

Output #3: 0

Output #4: 11193

Output #5: 95844

## American Computer Science League

---

### \* 1996–97 All-Star Contest \* 3. Digit Count – 10 Points \*

**Problem:** Consider those 5 digit numbers (i.e., between 10,000 and 99,999 inclusive) that can be written in the form

$$a^n b^m$$

where  $a$  and  $b$  are distinct prime numbers, and  $n$ , and  $m$  are non-negative integers. How many times does the digit  $k$  appear?

For example, there are four 5-digit numbers whose only prime factors are 13 and 19: 28561, 41743, 61009, and 89167. The digit zero appears twice, the digit one appears 4 times, and so on.

There will be 10 inputs, each consisting of three positive integers,  $a$ ,  $b$ , and  $k$ , in that order.

**Sample Input:**

Line #1: 5, 13, 6

Line #2: 11, 37, 2

...

**Sample Output:**

Output #1: 3

Output #2: 1

...

## American Computer Science League

---

\* 1996–97 All-Star Contest \* 3. Digit Count – 10 Points \*

**Test Input:**

Line #1: 7, 17, 1  
Line #2: 5, 23, 7  
Line #3: 11, 31, 1  
Line #4: 11, 19, 9  
Line #5: 17, 19, 9  
Line #6: 23, 7, 5  
Line #7: 2, 3, 6  
Line #8: 67, 37, 7  
Line #9: 19, 23, 9  
Line #10: 50051, 2, 5

**Test Output:**

Output #1: 8  
Output #2: 5  
Output #3: 7  
Output #4: 2  
Output #5: 1  
Output #6: 4  
Output #7: 23  
Output #8: 1  
Output #9: 0  
Output #10: 4

## American Computer Science League

---

### \* 1996–97 All-Star Contest \* 4. You Are El – 10/20 Points \*

**Problem:** Pages on the World Wide Web are accessed by a unique identifier called a URL (Uniform Resource Locator). For example, the URL of the file that you are reading right now is

`http://www.acsl.org/contests/96/ALLSTAR/prog4.ps`

There are three primary parts to a URL: protocol, host, and path. The *protocol*, `http` in this case, specifies how the file will be transferred from one computer to another. Other common protocols are `ftp`, `gopher`, and `file`. The *host* is the name of the computer on which the Web page resides. The ACSL Web pages all reside on the machine called `www.acsl.org`. Finally, the *path* (`contests/96/ALLSTAR/prog4.ps`) indicates where on the host computer the file is located.

The syntax of a URL is as follows: The protocol is the part of the URL up to the colon. The host is the part of the URL following the colon-slash-slash. It ends at a slash. The path starts after the slash ending the host. The path is optional; if it's missing, assume the path is `default.htm`. Also, if the path ends with a slash, append `default.htm` to it.

Links on a Web page *p* can be specified in one of three ways: in absolute terms, relative to the server hosting *p*, or relative to page *p*.

An absolute URL starts with a protocol (e.g., `http://www.cnn/`). This is used to jump to any arbitrary page on the Web. For example, if this page (the one you are reading right now!) contained the link `http://www.microsoft.com/`, clicking on it would cause you to jump to the Microsoft home page.

A relative-to-the-server URL starts with a slash (e.g., `/contests/95/SR/shorts3.ps`). This says to look for the page on the same server as *p*.

Finally, all other URLs are assumed to be relative to the page on which they reside. For example, the link `prog5.ps` on this page refers to the URL

`http://www.acsl.org/contests/96/ALLSTAR/prog5.ps`

The link `../INT/short2.ps` refers to `http://www.acsl.org/contests/96/INT/short2.ps`. That is, two dots indicates to go up a directory in the path. To make you life easy (see, we're nice guys!), the two dots will only appear at the start of a path. Of course, it might appear a few times (e.g., `../../94/JR/short1.ps`).

In this program, you'll be given the URL of a Web page and a link on the page, and you need to find the URL that the link will jump you to.

#### Sample Input:

Line #1: `http://www.acsl.org/contests/96/ALLSTAR/shorts.ps, programs.ps`  
Line #2: `http://www.acsl.org:8000/SampleContests/95/ALLSTAR/shorts.ps, /flyer.ps`  
Line #3: `http://www.acsl.org/SampleContests/95/ALLSTAR/shorts.ps, http://www.cnn/`  
...

#### Sample Output:

Output #1: `http://www.acsl.org/contests/96/ALLSTAR/programs.ps`  
Output #2: `http://www.acsl.org:8000/flyer.ps`  
Output #3: `http://www.cnn/default.htm`  
...

## American Computer Science League

---

\* 1996–97 All-Star Contest \* 4. You Are El – 10/20 Points \*

### Test Input:

Line #1: <http://www.acsl.org/contests/97/secret/shorts.ps>, [programs.ps](http://www.acsl.org/contests/97/secret/programs.ps)  
Line #2: <http://www.acsl.org/default.htm>, [source/main.html](http://www.acsl.org/source/main.html)  
Line #3: <http://www.acsl.org/home/front.html>, <http://www.xyz.com/>  
Line #4: <http://www.a.bb.ccc:1234/home.htm>, [xxx](http://www.a.bb.ccc:1234/home.htm)  
Line #5: <http://www.a.bb.ccc:1234/home.htm>, [xxx/](http://www.a.bb.ccc:1234/home.htm)  
Line #6: <http://gatekeeper.dec.com/pub/GNU/index.html>, [CVS/gcc.tar.gz](http://gatekeeper.dec.com/pub/GNU/ CVS/gcc.tar.gz)  
Line #7: <http://www.usgs.gov:251/nwis/bostom.htm>, <http://www.nasa.gov:8080/shuttle/>  
Line #8: <http://www.cs.state.edu/cs101/exams/final.htm>, [/cs321/intro.htm](http://www.cs.state.edu/cs321/intro.htm)  
Line #9: <http://www.cs.state.edu/cs101/exams/final.html>, [../homework/set1.ps](http://www.cs.state.edu/~/homework/set1.ps)  
Line #10: <http://www.cs.state.edu/cs101/exams/final.html>, [../~/ee543/missive.html](http://www.cs.state.edu/~/~/ee543/missive.html)

### Test Output:

Output #1: <http://www.acsl.org/contests/97/secret/programs.ps>  
Output #2: <http://www.acsl.org/source/main.html>  
Output #3: <http://www.xyz.com/default.htm>  
Output #4: <http://www.a.bb.ccc:1234/xxx>  
Output #5: <http://www.a.bb.ccc:1234/xxx/default.htm>  
Output #6: <http://gatekeeper.dec.com/pub/GNU/ CVS/gcc.tar.gz>  
Output #7: <http://www.nasa.gov:8080/shuttle/default.htm>  
Output #8: <http://www.cs.state.edu/cs321/intro.htm>  
Output #9: <http://www.cs.state.edu/cs101/homework/set1.ps>  
Output #10: <http://www.cs.state.edu/ee543/missive.html>

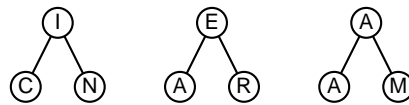
# American Computer Science League

## \* 1996–97 All-Star Contest \* 5. A Balancing Act, Revisited – 10 Points \*

**Problem:** In the last contest, you explored an approach to balancing a binary search tree: you maintained multiple binary search trees and as each new key was inserted, you inserted it into the tree where it would be closer to the tree's root. This problem tries a different approach at maintaining multiple trees.

The idea here is to consider the trees from left to right, and insert the new key into the first tree where it will be less than or equal to a specified depth  $k$ . If no such trees exist, start a new leftmost tree.

For example, inserting the string **AMERICANA** into trees of depth 1 or less will cause 3 trees to be created, whose roots are I, E, and A (in that order):



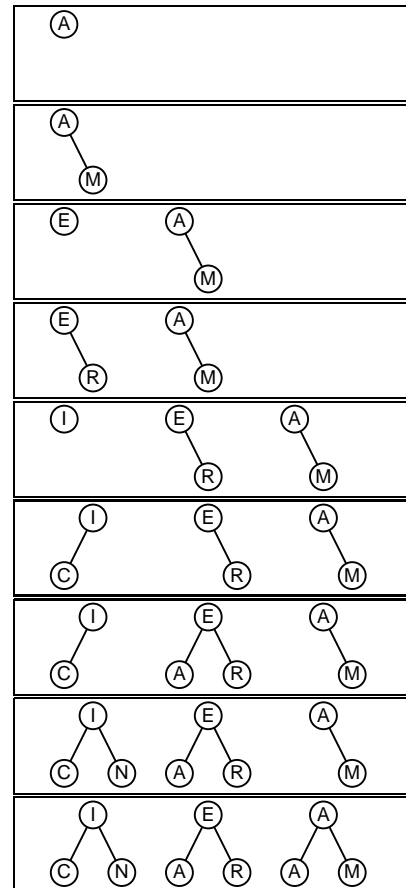
The input will be 10 sets of data. Each set will consist of a number  $k$  and a string  $s$  (up to 128 characters long). In the string  $s$ , ignore everything but the letters A through Z; uppercase and lowercase are the same. For each input pair, build the trees to depth  $k$  with the letters in  $s$  as described above. Print the root nodes of the trees in order from left to right.

**Sample Input:**

Line #1: 3, AMERICAN COMPUTER SCIENCE!  
 Line #2: 4, SALT LAKE CITY, UTAH  
 ...

**Sample Output:**

Output #1: E R C A  
 Output #2: E S  
 ...



## American Computer Science League

---

\* 1996–97 All-Star Contest \* 5. A Balancing Act, Revisited – 10 Points \*

### Test Input:

*Enter each as a single string, including punctuation.*

Line #1: 3, All Star Contest

Line #2: 5, THE WORLD WIDE WEB

Line #3: 4, ALL STAR CONTEST

Line #4: 4, I won first place in my division.

Line #5: 2, TIME FLIES LIKE AN ARROW. FRUIT FLIES LIKE A BANANA.

Line #6: 3, I MUST SAY THAT I FIND TELEVISION VERY EDUCATIONAL.

Line #7: 4, A Balancing Act

Line #8: 3, Bud: Strange as it may seem, they give ball players nowadays very peculiar names.

Line #9: 2, Bud: I'm telling you. Who's on first, What's on second, I Don't Know is on third.

Line #10: 0, computer

### Test Output:

Output #1: OA

Output #2: T

Output #3: NA

Output #4: ITI

Output #5: AKIRALFT

Output #6: NINTI

Output #7: GA

Output #8: EELEGTB

Output #9: ISNDSTOOLIB

Output #10: RETUPMOC



## American Computer Science League

---

### \* 1996–97 All-Star Contest \* 6. The Biggest, Revisited – 10 Points \*

**Problem:** You'd have thought that after Contest #1, you'd never hear from the Mighty Modulas again. But you'd be wrong. The Mighty Modulas have been at it again: They've been given index cards printed with numbers on both sides, and they've torn each card between the digits. Your job, as virtual leader of the Mighty Modulas, is to teach your charges how to tear each card so that the numbers on all the scraps add up as close as possible to some given target number without going over. You are free, of course, to turn over any scrap.

Consider, for example, a card with the number 534 on one side and 197 on the other side. There are 3 ways that you could tear the card: between the 5 and the 3; between the 3 and the 4; and after the 5 and after the 3. In the first case, you have two strips: one with 5 on one side and 7 on the other, and the other strip with 34 on one side and 19 on the other. In the second case, you'd also have two strips: one with 53 on one side and 97 on the other, and the second strip with 4 on one side and 1 on the other. Finally, in the third case, you'd have 3 strips. You must make at least one tear. If the numbers are different lengths, pad the shorter one with leading zeros so that they are of the same length.

You'll be given 10 sets of data. Each set consists of three numbers: the number on one side of the card, the number on other side of the card, and a target number. You are to figure out how to tear the card such that the numbers on one side or the other of the torn scraps add up as close as possible to the target number without going over. Print out the sum. The number on the cards will be positive integers less than 1,000,000,000.

**Sample Input:**

Line #1: 534, 197, 50  
Line #2: 534, 197, 100  
Line #3: 251, 8, 60  
...

**Sample Output:**

Output #1: 41  
Output #2: 98  
Output #3: 59  
...

## American Computer Science League

---

\* 1996–97 All-Star Contest \* 6. The Biggest, Revisited – 10 Points \*

**Test Input:**

Line #1: 23, 45, 10  
Line #2: 23, 45, 30  
Line #3: 534, 197, 200  
Line #4: 1234, 1234, 15  
Line #5: 1234, 1234, 60  
Line #6: 1234, 1234, 400  
Line #7: 23, 45678, 3000  
Line #8: 987654, 111, 200  
Line #9: 13579, 2468, 3000  
Line #10: 232456, 813579, 25000

**Test Output:**

Output #1: 9  
Output #2: 9  
Output #3: 101  
Output #4: 15  
Output #5: 46  
Output #6: 238  
Output #7: 723  
Output #8: 183  
Output #9: 2477  
Output #10: 23253

# American Computer Science League

---

## \* 1996–97 All-Star Contest \* 7. Squarepoint – 10 Points \*

**Problem:** Given a rectangle and a point, report the distance from the point to the nearest point on the perimeter of the rectangle.

There will be ten sets of data. Each set of data consists of the two corners of a rectangle, followed by coordinates  $(x, y)$  of a point  $p$ . To make input easy on the proctor, the corners of the rectangle will be a character that refers to one of the following 9 points:

|           |           |           |
|-----------|-----------|-----------|
| A (2, 3)  | B (11, 6) | C (7, 11) |
| D (1, 10) | E (5, 16) | F (5, 6)  |
| G (0, 40) | H (9, 9)  | I (7, 0)  |

For each input, print the distance from  $p$  to the closest spot on the perimeter of the rectangle. Your answers must be within 0.01 of our answers

**Sample Input:**

Line #1: F, H, 7, 8

Line #2: C, B, 5, 9

...

**Sample Output:**

Output #1: 1

Output #2: 2

...

# American Computer Science League

---

\* 1996–97 All-Star Contest \* 7. Squarepoint – 10 Points \*

**Test Input:**

Line #1: A, B, 5, 5  
Line #2: A, B, 0, 6  
Line #3: B, A, 9, 0  
Line #4: B, A, 15, 6  
Line #5: B, A, 14, 10  
Line #6: C, F, 10, 2  
Line #7: C, F, 7, 7  
Line #8: F, C, 5, 5  
Line #9: F, C, 12, 23  
Line #10: F, C, 5, 6

**Test Output:**

*Answers must match within 0.01*

Output #1: 1  
Output #2: 2  
Output #3: 3  
Output #4: 4  
Output #5: 5  
Output #6: 5  
Output #7: 0  
Output #8: 1  
Output #9: 13  
Output #10: 0

# American Computer Science League

## \* 1996–97 All-Star Contest \* 8. Alphaddition – 10 Points \*

**Problem:** Remember those “brain puzzlers” of the form:

$$ABCD + BDC = EAEA$$

In case you don't remember them, the problem is pretty straightforward: find numbers to replace the A, B, C, . . . that make the equation true. The replacements must be consistent; that is, replace all A's by one digit, all B's by a different digit, and so on. For example, the equation  $1538 + 583 = 2121$  solves the equation above. The equation has 5 more solutions:  $1547 + 574 = 2121$ ,  $1574 + 547 = 2121$ ,  $1583 + 538 = 2121$ ,  $3658 + 685 = 4343$ , and  $3685 + 658 = 4343$ . The second sample problem below has three solutions:  $546 + 54 = 600$ ,  $576 + 57 = 633$ , and  $586 + 58 = 644$ .

There will be 10 inputs. Each input consists of three strings,  $p$ ,  $q$ , and  $r$ . The strings are composed of the letters A through Z and numbers. You need to report the number of solutions there are to the problem  $p + q = r$ . The leading digit in each term cannot be a 0. Also, to keep the running time of this program reasonable, we promise that there won't be more than 4 different letters in the original data.

**Sample Input:**

Line #1: AA + BC = 100  
Line #2: XYZ + XY = 6QQ  
...

**Sample Output:**

Output #1: 7 (solutions are: 11+89, 22+78, 33+67, 44+56, 66+34, 77+23, 88+12)  
Output #2: 3 (solutions are: 546+54, 576+57, 586+58)  
...

# American Computer Science League

---

\* 1996–97 All-Star Contest \* 8. Alphaddition – 10 Points \*

**Test Input:**

Line #1:  $1996 + ABBC = 3993$

Line #2:  $1996 + ABBC = D99D$

Line #3:  $AB + ABC = 6DD$

Line #4:  $4BCD + 4BCD = 8FF8$

Line #5:  $3BBCC + 58DC3 = CDEC2$

Line #6:  $PQRS + RS = PQQP$

Line #7:  $ABBC + 779 = CE2A$

Line #8:  $AA6C5 + 62AC = ACD9F$

Line #9:  $XYXX + XX = XYZZ$

Line #10:  $A + 77DE = 77F7$

**Test Output:**

Output #1: 1

Output #2: 5

Output #3: 3

Output #4: 4

Output #5: 7

Output #6: 20

Output #7: 4

Output #8: 1

Output #9: 32

Output #10: 14