Midwest Instruction & Computing Symposium (MICS)
University of Wisconsin-La Crosse
April 19th – 20th, 2013

Student Programming Contest Problem Set

(This page is intentionally left blank)

Problem 1—Twin Prime Pairs

Professor Plum likes to keep his elementary-aged childrens' minds active over the summer months with simple mathematical puzzles. He asked them to find all twin prime pairs between two given positive integers.

A *twin prime* is a prime number that differs from another prime number by 2. For example a twin prime pair is (41, 43) since both 41 and 43 are prime numbers.

The first few twin prime pairs are:
(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139), …

INPUT SPECIFICATION

The only line of input will contain two positive integers separated by a single blank. The first integer will be the smaller one, and the second integer will fit in a 32-bit binary representation.

OUTPUT SPECIFICATION

This problem should produce all twin prime pairs entirely within the input values inclusive of these values. Each pair should be on a line by themselves and be of the form `(#, #)` with the smaller prime being listed first. Note the primes are enclosed by parentheses, and separated by a comma and a single blank space.

SAMPLE INPUT

```
11 72
```

SAMPLE OUTPUT

```
(11, 13)
(17, 19)
(29, 31)
(41, 43)
(59, 61)
```

(This page is intentionally left blank)

Problem 2—Alternate Sort

Unlike most old professors, Professor Plum likes change. While writing an array question for his final examination in CS 101, he invents the notion of an *alternate sort* where the smallest item is at the first index, the largest item is at the second index, the second smallest item is at the third index, the second largest item is at the fourth index, etc. All integers will fit in a 32-bit binary representation.

For example, an array initially ordered as: 20, 45, 30, 5, 15, 50, 10, 30 would be alternate sorted to: 5, 50, 10, 45, 15, 30, 20, 30.

INPUT SPECIFICATION

The input contains the array to be sorted, one integer per line.

OUTPUT SPECIFICATION

The output should contain the sorted array (in alternate sorted order), one integer per line.

SAMPLE INPUT

```
20
45
30
5
15
50
10
30
```

SAMPLE OUTPUT

```
5
50
10
45
15
30
20
30
```

(This page is intentionally left blank)

Problem 3—GIF Decompression

Professor Plum enjoys telling stories to students about his adventures in industry before he joined the faculty. In 1987 he was working at CompuServe when the Graphics Interchange Format (GIF) encoding was created to compress image files. This was an inspiring event is his life, but lately he has been having trouble remembering how the decompression algorithm works and has asked you to help him.

In this problem, we will consider a simplified version of the GIF encoding algorithm applied to strings of alphabetic characters. Essential for this compression is a dictionary that assigns numeric encodings (we will use base 10 numbers for this problem) to different strings of characters. The dictionary is initialized with mappings for characters or substrings which may appear in the string. For example, if we expect to encounter all 26 letters of the alphabet, the dictionary will initially store the encodings (`A,00`), (`B,01`), (`C,02`), …, (`Z,25`). If we are compressing DNA data, the dictionary will initially store only 4 entries: (`A,0`), (`T,1`), (`G,2`) and (`C,3`). Note that the length of each initial encoding is the same for all entries (2 digits in the first example, and 1 digit in the second).

The compression algorithm proceeds as follows:

1. Find the longest prefix of the uncompressed portion of the string which is in the dictionary, and replace it with its numeric encoding.
2. If the end of the string has not been reached, add a new mapping **(*s*,*n*)** to the dictionary, where ***s*** = the prefix just compressed plus the next character after it in the string, and ***n*** = the smallest number not yet used in the dictionary.

For example, assume we started with the string `ABABBAABB` and a dictionary with just two entries, (`A,0`) and (`B,1`). The table below shows the steps in compressing the string.

String	Longest Prefix	Replaced With	New Dictionary Entry
ABABBAABB	A	0	(AB,2)
0BABBAABB	B	1	(BA,3)
01ABBAABB	AB	2	(ABB,4)
012BAABB	BA	3	(BAA,5)
0123ABB	ABB	4	-

The final compressed string is `01234`.

There is only one other rule: the replacement strings used are always the size of the longest encoding in the dictionary at the time the replacement occurs. Thus, with the dictionary above, if the string to compress is long enough that an entry of the form (*s*, `10`) is added to the dictionary, then from this point on all numerical replacement strings used in the compressed string must be expanded to 2 digits long (i.e., `A` will now be encoded as `00`, `B` as `01`, `AB` as `02`, etc.); if an entry (*s'*, `100`) is added to the dictionary, all replacements from this point forward will increase to 3 digits long, and so on. Thus, the longer string `ABABBAABBAABAABAB` will be encoded as `01234027301`, not `0123402731`. Try it!

OK, now that you are experts at compressing, it's time to relax and decompress!

INPUT SPECIFICATION

Each test case will consist of two lines. The first line will contain a string of digits to decompress. The second line will contain the initial dictionary used in the compression. This line will start with a positive integer ***n*** indicating the number of entries in the dictionary ($1 \leq n \leq 100$), followed by ***n*** alphabetic strings. The first of these will be paired with 0 in the dictionary (or 00 if n > 10), the second with 1, and so on. The last test case will be followed by a line containing a single 0.

OUTPUT SPECIFICATION

For each test case, output a single line containing the word `Case`, a single space, the case number (starting at 1), a colon, a single space, and the decompressed string. All input strings will have been legally compressed.

SAMPLE INPUT

```
01234
2 A B
01234027301
2 A B
02151120182729
26 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
21104
3 BA A C
01
2 JA VA
0
```

SAMPLE OUTPUT

```
Case 1: ABABBAABB
Case 2: ABABBAABBAABAABAB
Case 3: CPLUSPLUS
Case 4: CAABAAA
Case 5: JAVA
```

Problem 4—Farkel Roll

Professor Plum likes to play the dice game Farkel, so he is trying to teach a Computer Science colleague to play.

A player's turn consists of an initial roll of 6 six-sided dice. The player removes the dice they want to use for points on that roll. The player can decide to stop or reroll the remaining dice. If the player rerolls the remaining dice and does not score any points, then the player loses all their points for that turn (called a *Farkel*). If the player stops, the points for all the rolls in that turn are added to their total score.

Professor Plum wants you to develop a training program for scoring a single (re)roll within a turn. The (re)roll could consist of between 1 to 6 dice with the program returning the maximum score for the (re)rolled dice. In Farkel scoring only ONES and FIVES can count individually, but other dice can count in the following combinations:

ONES are 100 points each	FIVES are 50 points each
3 ONES are 300 points	4 of any kind is 1000 points
3 TWOS are 200 points	5 of any kind is 2000 points
3 THREES are 300 points	6 of any kind is 3000 points
3 FOURS are 400 points	Straight ONE to SIX is 1500 points
3 FIVES are 500 points	Three pairs is 1500 points
3 SIXES are 600 points	Two Triplets is 2500 points

Some sample (re)rolls and their corresponding scores are:

(Re)roll Dice	Score	Comments
1 1 3 5	250	Two ONES (2 x 100) plus a single FIVE (1 x 50)
1 5 5 5 6	600	Single ONE (100) plus 3 FIVES for 500
2 2 3 4 6 6	0	Nothing counts for points – this is called a "Farkel"
2 2 2 2 4 5	1050	4 TWOS (1000) plus a single FIVE for 50
5 6	50	Single FIVE for 50
1	100	Single ONE for 100

INPUT SPECIFICATION

Each line of input contains between 1 and 6 dice values for a single (re)roll separated by whitespace. **The dice values on a line are NOT sorted as in the above examples.**

OUTPUT SPECIFICATION

The output should contain one line of output corresponding to each line of input. Each output line echos the dice values sorted in ascending order, an equal sign (" = ") with a single space on each side, and the score for the (re)roll.

SAMPLE INPUT

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

SAMPLE OUTPUT

```
1 1 3 5 = 250
5 6 = 50
2 2 3 4 6 6 = 0
5 = 50
```

Problem 5—Number of Admissible Paths

Professor Plum likes to vacation in big cities because he can relate to the "squareness" of the city blocks. The city blocks remind him of a two-dimensional lattice.

Let's call a lattice point (x, y) *inadmissible* if x, y and $x + y$ are all positive perfect squares.
For example, (9, 16) is inadmissible, while (0, 4), (3, 1) and (9, 4) are not.

Consider a path from point (x_1, y_1) to point (x_2, y_2) using only unit steps north or east.
Let's call such a path *admissible* if none of its intermediate points are inadmissible.

Let P(n) be the number of admissible paths from (0, 0) to (n, n).
It can be verified that P(3) = 20, P(5) = 252, and P(16) = 596994440.

INPUT SPECIFICATION
The input contains a single line with a positive integer n. The resulting P(n) value will fit in a 64-bit integer representation.

OUTPUT SPECIFICATION
The output should contain a single line with the number of admissible paths from (0, 0) to (n, n).

SAMPLE INPUT
```
16
```

SAMPLE OUTPUT
```
596994440
```

(This page is intentionally left blank)

Problem 6—Alien Math

On Saturday nights, Professor Plum takes off his aluminum-foil helmet and tunes his FM radio between two local stations. He is convinced that aliens are broadcasting on that frequency. Somehow he has deducted from listening to the static that the aliens have 4-fingers on each of 3 hands. To prepare for the alien invasion, he wants to practice his base-12 mathematics. He decided to use the digits: 0 – 9, A, and B, and focus only on addition.

Professor Plum wants you to develop a base-12 addition program to check his answers.

INPUT SPECIFICATION

Each line of input contains two base-12 numbers separated by a single space. Each base-12 number will contain at most 50 digits.

OUTPUT SPECIFICATION

The output file should contain one line of output corresponding to each line of input. Each output line contains the corresponding base-12 sum.

SAMPLE INPUT

```
934A5 349
546A 96AB7
55555555555555555555 55555555555555555555
6BAA0A5 4482
```

SAMPLE OUTPUT

```
93832
A0365
AAAAAAAAAAAAAAAAAAAA
6BB2567
```

(This page is intentionally left blank)

Problem 7—Custom Customer Mailings

Professor Plum's wife runs a small mail-order business. She has a spreadsheet of customer data in a .csv (comma separated values) text-file. The first line contains comma separated column headings with the remaining lines containing customer data one-per-line. The first few lines of this file are:

```
FIRST,MI,LAST,STREETADDRESS,CITY,STATECODE,ZIP,COUNTRY,EMAIL,PHONE,GENDER,BIRTHDAY
Woodrow,C,Wilson,2362 New Street,Eugene,OR,97408,US,Woodrow.C.Wilson@spambob.com,541-337-9453,male,11/26/1984
Eric,A,Stutler,568 Nuzum Court,East Aurora,NY,14052,US,Eric.A.Stutler@trashymail.com,716-652-4943,male,11/24/1947
David,E,Herbert,3678 Seltice Way,Boise,ID,83704,US,David.E.Herbert@spambob.com,208-703-8246,male,2/16/1971
Lee,A,Andrews,3472 Berkshire Circle,Knoxville,TN,37917,US,Lee.A.Andrews@spambob.com,865-566-4125,male,11/15/1973
Rena,D,Adkins,3153 Cardinal Lane,Cleveland Heights,OH,44118,US,Rena.D.Adkins@trashymail.com,216-932-7637,female,1/14/1975
Reyna,R,Brown,2093 Middleville Road,Los Angeles,CA,90014,US,Reyna.R.Brown@pookmail.com,626-388-0030,female,3/20/1942
Duane,E,Hall,3467 Ray Court,WAGRAM,NC,28396,US,Duane.E.Hall@pookmail.com,910-369-3902,male,7/30/1962
…
```

His wife wants to be able to generate custom customer mailing labels which target customers matching a specified query. The query components can be based on gender, state, and/or decade born. A sample query would be:

```
GENDER=male
STATECODE=OR
STATECODE=ID
DECADE=1970
DECADE=1980
DECADE=1960
```

This query's meaning would be all males living in Oregon or Idaho born in the 1960s, 1970s, or 1980s.

The output file should contain a mailing label for each customer satisfying the query. Each mailing label should be in **all upper-case letters** and consist of:

- one blank line to start
- a name line with the first name, a blank space, the last name
- street address line
- the city, a comma, a blank space, the state code, **two blank spaces**, the zip code
- end with a blank line

For the data visible above, the above query would produce the mailing labels:

```
WOODROW WILSON
2362 NEW STREET
EUGENE, OR  97408


DAVID HERBERT
3678 SELTICE WAY
BOISE, ID  83704
```

INPUT SPECIFICATION

The input will start with a query (one or more lines for the query), **a blank line**, and finally the customer (.csv) data.

OUTPUT SPECIFICATION

The output should contain the mailing labels for each customer satisfying the query. The mailing labels should be in the same order that the corresponding customers appeared in the (.csv) data.

SAMPLE INPUT

```
GENDER=female
GENDER=male
STATECODE=MN
STATECODE=IA
DECADE=1980
DECADE=1950

FIRST,MI,LAST,STREETADDRESS,CITY,STATECODE,ZIP,COUNTRY,EMAIL,PHONE,GENDER,BIRTHDAY
Woodrow,C,Wilson,2362 New Street,Eugene,OR,97408,US,Woodrow.C.Wilson@spambob.com,541-337-9453,male,11/26/1984
Eric,A,Stutler,568 Nuzum Court,East Aurora,NY,14052,US,Eric.A.Stutler@trashymail.com,716-652-4943,male,11/24/1947
David,E,Herbert,3678 Seltice Way,Boise,ID,83704,US,David.E.Herbert@spambob.com,208-703-8246,male,2/16/1971
Lee,A,Andrews,3472 Berkshire Circle,Knoxville,TN,37917,US,Lee.A.Andrews@spambob.com,865-566-4125,male,11/15/1973
Rena,D,Adkins,3153 Cardinal Lane,Cleveland Heights,OH,44118,US,Rena.D.Adkins@trashymail.com,216-932-7637,female,1/14/1975
Reyna,R,Brown,2093 Middleville Road,Los Angeles,CA,90014,US,Reyna.R.Brown@pookmail.com,626-388-0030,female,3/20/1942
Duane,E,Hall,3467 Ray Court,WAGRAM,NC,28396,US,Duane.E.Hall@pookmail.com,910-369-3902,male,7/30/1962
Fannie,C,Chapman,1791 Shingleton Road,Bridgman,MI,49106,US,Fannie.C.Chapman@spambob.com,269-465-7477,female,10/5/1961
Maggie,D,Betts,1972 Twin Willow Lane,Fayetteville,NC,28304,US,Maggie.D.Betts@spambob.com,910-425-9414,female,8/16/1980
Glenn,M,Stephens,280 Hillcrest Circle,MAPLE PLAIN,MN,55359,US,Glenn.M.Stephens@dodgit.com,763-479-7845,male,7/18/1985
Eric,T,Sanders,4624 Broadway Street,BLUFFTON,SC,29910,US,Eric.T.Sanders@dodgit.com,843-705-4450,male,9/4/1951
Stephen,L,Mcknight,4078 White River Way,West Valley City,UT,84119,US,Stephen.L.Mcknight@spambob.com,801-515-8270,male,7/31/1962
Javier,A,Powers,4920 Lowndes Hill Park Road,Sherman Oaks,CA,91403,US,Javier.A.Powers@dodgit.com,661-351-8863,male,7/5/1982
Diana,A,Russell,250 Oral Lake Road,NEW MARKET,MN,55054,US,Diana.A.Russell@spambob.com,952-461-5925,female,5/14/1953
Donna,D,Goodrich,4977 Bastin Drive,Eagleville,PA,19403,US,Donna.D.Goodrich@mailinator.com,484-764-2478,female,4/6/1971
Jess,H,Franklin,585 Ashford Drive,Mc Lean,VA,22102,US,Jess.H.Franklin@trashymail.com,703-667-1447,male,9/19/1943
Jason,H,Morin,929 Village View Drive,Fort Myers,FL,33901,US,Jason.H.Morin@pookmail.com,239-846-4951,male,4/26/1970
Rose,L,Rice,492 Leroy Lane,Watertown,SD,57201,US,Rose.L.Rice@mailinator.com,605-882-6217,female,2/18/1985
Julia,D,Goble,885 Cherry Camp Road,Chicago,IL,60605,US,Julia.D.Goble@mailinator.com,773-532-8122,female,7/5/1952
James,D,Drew,4355 Buckhannan Avenue,Syracuse,NY,13202,US,James.D.Drew@spambob.com,315-474-1613,male,4/4/1945
Duane,A,Aviles,818 Court Street,Saint Louis,MO,63146,US,Duane.A.Aviles@trashymail.com,636-893-4862,male,5/22/1973
Robert,K,Mullens,3188 Short Street,Georgetown,TX,78626,US,Robert.K.Mullens@mailinator.com,512-942-2285,male,10/7/1960
Daniel,V,Hearn,1811 Red Hawk Road,MARIETTA,MN,56257,US,Daniel.V.Hearn@mailinator.com,320-668-8346,male,9/19/1982
Billy,V,Thompson,2095 Mesa Drive,LAS VEGAS,NV,89128,US,Billy.V.Thompson@spambob.com,702-341-5879,male,5/5/1961
Daniel,D,King,3916 Lodgeville Road,Golden Valley,MN,55427,US,Daniel.D.King@mailinator.com,612-269-7558,male,5/31/1967
Demetrius,C,Fu,1321 Collins Street,Tampa,FL,33634,US,Demetrius.C.Fu@pookmail.com,813-841-2434,male,3/25/1985
Clarence,M,Kohler,3638 Marshville Road,Garden City,NY,11530,US,Clarence.M.Kohler@pookmail.com,845-664-2670,male,6/14/1957
Salvatore,M,Salisbury,395 Hornor Avenue,Tulsa,OK,74120,US,Salvatore.M.Salisbury@pookmail.com,918-329-7031,male,8/14/1968
Mildred,E,Wade,4760 Layman Avenue,Wilmington,NC,28403,US,Mildred.E.Wade@pookmail.com,910-933-6397,female,6/8/1944
Rosie,E,Gaunt,4174 Joyce Street,LILLIAN,AL,36542,US,Rosie.E.Gaunt@pookmail.com,251-962-0105,female,1/11/1981
James,M,Wood,2701 Goldleaf Lane,JERSEY CITY,NJ,07304,US,James.M.Wood@mailinator.com,201-631-5596,male,11/29/1968
Dolores,J,Charon,4552 Hillside Street,MESA,AZ,85225,US,Dolores.J.Charon@pookmail.com,480-503-3567,female,8/31/1967
Cora,M,Mcculloch,3462 Mill Street,Saint Petersburg,FL,33716,US,Cora.M.Mcculloch@pookmail.com,863-981-1677,female,8/30/1958
Peter,N,Underwood,2144 Kildeer Drive,Newport News,VA,23601,US,Peter.N.Underwood@spambob.com,757-240-5459,male,5/15/1961
Bobbie,J,Quintero,1820 Quincy Street,Philadelphia,PA,19107,US,Bobbie.J.Quintero@trashymail.com,267-555-1272,female,10/3/1977
```

SAMPLE OUTPUT

```
<EOLN>
GLENN STEPHENS<EOLN>
280 HILLCREST CIRCLE<EOLN>
MAPLE PLAIN, MN  55359<EOLN>
<EOLN>
<EOLN>
DIANA RUSSELL<EOLN>
250 ORAL LAKE ROAD<EOLN>
NEW MARKET, MN  55054<EOLN>
<EOLN>
<EOLN>
DANIEL HEARN<EOLN>
1811 RED HAWK ROAD<EOLN>
MARIETTA, MN  56257<EOLN>
<EOLN>
<EOF>
```

←The **<EOLN>** tag represents the end-of-line, and **should not be printed**.
←The **<EOF>** tag represents the end-of-file, and **should not be printed**.