# New Zealand Programming Contest 2007


## Preamble

Please note the following very important details:

1) Read all input from the keyboard, i.e. use `stdin`, `System.in`, `cin` or equivalent. Input will be redirected from a file to form the input to your submission.

2) Write all output to the screen, i.e. use `stdout`, `System.out`, `cout` or equivalent. Do not write to `stderr`. Do NOT use, or even include, any module that allows direct manipulation of the screen, such as `conio`, `Crt` or anything similar. Output from your program is redirected to a file for later checking. Use of direct I/O means that such output is not redirected and hence cannot be checked. This could mean that a correct program is rejected!

3) Unless otherwise stated, all *integers* in the input will fit into a standard 32-bit computer word. Adjacent integers on a line will be separated by one or more spaces.

4) An *uppercase letter* is a character in the sequence 'A' to 'Z', a *lowercase letter* is a character in the sequence 'a' to 'z', and a *letter* is either.

5) *Whitespace* means any or all of blanks, tabs, and the starts and ends of lines.

6) If it is stated that 'a line contains no more than *n* characters', this does not include the character(s) specifying the end of line.

7) A note on leap years. In the Gregorian calendar in use in New Zealand,
   if a year is divisible by 4 then it is a leap year, **unless**
   it is divisible by 100, in which case it isn't, **unless**
   it is divisible by 400, in which case it is.
   Thus 1976 and 2000 are leap years, but 1900 is not. In a leap year, February has 29 days, otherwise it has 28.

# New Zealand Programming Contest 2007

## Problem A Coin tossing 3 points

When I was at school, many, many years ago, we used to play a simple game involving tossing a coin. The first player would call "heads" or "tails", the second would toss the coin. The first player gained a point for every correct call, the second player gained one for every incorrect call. When we got bored, we added up the scores!

Input for this problem will consist of a number of games. Each game will begin with a line containing 2 names of no more than 20 letters each and separated by a space. The second line of each game will contain a single positive integer, *n*, specifying the number of recorded coin tosses (0 < n <= 1000). *n* coin tosses follow, each on a single lines containing 2 characters (either 'H' or 'T' – "heads" or "tails"), separated by a space. The first character denotes the call and the second denotes the result. Input will be terminated by a line containing just # #.

Output will be one line for each game. The line will give the name of the first player (as recorded in the input), followed by the first player's score, followed by the name of the second player, followed by the second player's score. Entries on a line will be separated by a single space.

### Sample Input

```
Sally John
5
H H
H T
H H
T T
T H
Eloise Ahmed
4
H T
T T
H H
T H
# #
```

### Sample Output

```
Sally 3 John 2
Eloise 2 Ahmed 2
```

# New Zealand Programming Contest 2007

## Problem B  Have you had your birthday yet?  3 points

Today it is 4th August. If you were born before 4th August (in whatever year you were born) then you have already had your 2007 birthday. If you were born after 4th August, you have not yet had your 2007 birthday. If you were born on 4th August, happy birthday! If you were born on 29th February unlucky, you do not have a birthday this year, as 2007 is not a leap year!

Input will consist of a number of lines containing dates. All dates will be valid and no date will be repeated, so there will be no more than 365 lines. The format for a line will be one or two digits, a space, and the full name of a month. The name will begin with an upper case letter, and the other letters will be lower case. Input will be terminated by a line containing just 0 #.

Output will consist of the word "Yes" if a person born on that date would have had their 2007 birthday by now, "No" if they would not, "Happy birthday" if the date is 4th August and "Unlucky" if the date is 29th February. The first letter on each output line is upper case, the rest are lower case.

### Sample Input

```
5 January
15 December
29 February
4 August
8 August
0 #
```

### Sample Output

```
Yes
No
Unlucky
Happy birthday
No
```

# New Zealand Programming Contest 2007

## Problem C Hide those Letters 3 points

Mrs Jones, a primary school teacher, has thought of a way to help her young pupils to learn to spell. She is removing certain letters from sentences and replacing them with underscore characters. The children have to write the appropriate letters in the spaces. As a starting point, Mrs Jones takes 2 different letters from a sentence (such as a and e), and the children have to work out whether each underscore represents an 'a' or an 'e', for example.

Input to this problem will consist of a number of cases, terminated by a line containing just # #. Each case will begin with two lower case letters on a single line, separated by a space, followed by a line containing a single integer $n$ ($1 \le n \le 100$) which gives the number of lines of text making up the case. The following $n$ lines each contain a line of text of up to 255 characters.

Output for each case will be the case number, followed by 1 line for each line of text presented in the input. The text will be as input but with every occurrence of either of the two letters replaced by an underscore character (_). Although the letters provided are in lowercase, uppercase versions of the same letter must also be replaced. Leave a blank line between cases.

### Sample Input

```
a e
2
Come here Evans, said the teacher.
I think not.
p r
1
Purple reindeer rarely appear in Ruritania.
# #
```

### Sample Output

```
Case 1
Com_ h_r_ _v_ns, s_id th_ t__ch_r.
I think not.

Case 2
_u__le _eindee_ _a_ely a__ea_ in _u_itania.
```

# New Zealand Programming Contest 2007

## Problem D Rectangles 3 points

My young son is struggling with his maths homework, can you help him please? He is working on the area of a rectangle – you know, area = length x width. His teacher has given him a table of lengths, widths and areas. Each row on the table has one of the 3 values missing; my son has to work out the missing value and write it in the table such that the values on each line represent the length, width and area of one rectangle.

Input is a series of lines, each containing 3 integers, *l*, *w*, and *a* ( $0 \le l\ w \le 100$, $0 \le a \le 10{,}000$) representing the length, width and area of a rectangle in that order. The integers are separated by a single space. In each row, only one of the values has been replaced by a zero. The final row contains 0 0 0 and should not be processed.

Output is the same series of lines but with the zero in each line replaced by the correct value for the length, width or area as appropriate. The new value is always an integer.

### Sample Input

```
2 0 6
6 5 0
0 8 80
9 0 45
0 0 0
```

### Sample Output

```
2 3 6
6 5 30
10 8 80
9 5 45
```

# New Zealand Programming Contest 2007

## Problem E Word Extraction 10 points

I am trying to create a dictionary of all the words in common use by my students. To do this, I am planning to feed text from their work, discussion forum entries, and emails through a program that extracts the words.

To make it easy to check if a word has been seen before, I am formatting the text carefully. Firstly I put everything in lower case, then I strip out all punctuation leaving the words separated by single spaces. If the punctuation comes within a word (i.e. there is a letter on either side of it), it is removed. If a "word" consists only of digits, it is ignored. Finally, I sort the words into alphabetical order, removing duplicates.

Input will consist of a number of lines, each line representing a piece of text to be analysed, and terminated by a single #. No line will contain more than 250 characters.

Output will be the qualifying words extracted from the text, one per line, in alphabetical order within each set. Each set of words from a piece of text will be separated from the next set by a blank line.

**Sample input** (Note that line breaks are denoted by '<eol>' because of the narrow page, '<eol>' does **not** appear in the actual input.)

```
Can I please have the spec for the Programming 3 assign-
ment?  B.T.W. I haven't got the lecture notes either!<eol>
Our guest speaker is Mr Hartley-Jones.  He shouldn't need
any introduction - you all met him last semester, didn't
you?<eol>
When you said "give me your best 5 examples", did you mean
just from this year - or can we use any?  I'd like to use
one from2006.<eol>
#
```

**Sample output**

(This is arranged in 4 columns to save space)

```
assignment      all             speaker         one
btw             any             you             or
can             didnt                           said
either          guest           any             this
for             hartleyjones    best            to
got             he              can             use
have            him             did             we
havent          introduction    examples        when
i               is              from            year
lecture         last            from2006        you
notes           met             give            your
please          mr              id
programming     need            just
spec            our             like
the             semester        me
                shouldnt        mean
```

# New Zealand Programming Contest 2007

## Problem F Bree's pantry 10 points

Those who watch "Desperate Housewives" will have encountered Bree, who is an obsessive compulsive. She likes to keep her cans in order in her pantry — tallest in the middle, next tallest to the left of the middle one, next tallest to the right of the middle one and so on. Her tins are all labelled with a 3 letter code to show their contents, such as "TOM" for tomatoes, or "swc" for sweet corn. If two or more cans have equal heights, she considers cans that are labelled the lowest alphabetically (ignoring case) to be the tallest.

Input consists of a series of scenarios, terminated by a line containing a single zero (0). Each scenario begins with a line containing the number of cans to be arranged ($n$, $1 \le n \le 20$). This line is followed by $n$ lines each representing one can, denoted by a 3 letter code, a space, and a positive integer specifying the height of the can in centimetres.

Output for each scenario will be a single line, the codes from the cans in the order, from left to right, that they will appear in Bree's pantry. The case for each code must be the same as it was on input.

### Sample input

```
5
TOM 12
Veg 10
SPG 15
XXX 9
swc 10
3
Alm 8
SAL 6
Tna 5
0
```

### Sample output

```
Veg TOM SPG swc XXX
SAL Alm Tna
```

# New Zealand Programming Contest 2007

## Problem G Paula's search 10 points

Paula is searching for the perfect pink ball dress. In her High Street are 100 shops spread evenly along both sides of the street, odd numbers on one side, even numbers on the other as usual.

Paula always begins her search by going into the nearest shop to her home, which is shop 50, right in the middle of the even numbered side of the road. The shop owners all know where her dress of choice is, but they like to offer Paula a challenge, so they have agreed to tell her only one of 4 things:

1. Yes, I have the perfect dress for you!
2. You are on the wrong side of the road, cross over.
3. You need to look down this side of the street (amongst the lower numbered shops).
4. You need to look up this side of the street (amongst the higher numbered shops).

If Paula gets answer 1, she buys the dress and goes home. If she gets answers 2, 3 or 4, Paula always goes to the mid point of the block of shops in which she has to search (if there is an even number of shops, she goes to the lower numbered of the two middle shops). Here the shop owner gives her one of the 4 responses and Paula continues her search if necessary.

Input will be a number of scenarios, each one being a single integer (between 1 and 100 inclusive) on a line on its own representing the number of the shop where Paula will find her perfect pink ball dress. Input will be terminated by a single 0, this line should not be processed.

Output will be a single integer (on a line on its own) for each line of input. The integer will be the number of shops Paula had to visit in order to buy her dress.

### Sample input

```
15
74
50
0
```

### Sample output

```
7
6
1
```

# New Zealand Programming Contest 2007

## Problem H Cedric's Cypher 10 points

A Caesar cipher is a way of encrypting text. Each letter in the text is replaced by the letter *s* places after it in the alphabet. *S* is called the shift – with a shift of 5, for example, 'A' becomes 'F', 'H' becomes 'M' and 'Y' becomes 'D '(the alphabet wraps round so 'A' follows 'Z'). The text is easy to decrypt if you know the value of *s*.

Cedric has come up with a variation on this. Instead of telling his friends what the shift is, he encrypts the letter 'A' (always producing an upper case letter) and places it at the end of his message. In that way, they can work out what the shift is and thus decrypt the text.

Input will consist of a number of lines of text, terminated by a line consisting of only #. Each line will consist of up to 255 characters of encrypted text (including the encrypted 'A').

Output will be one line for each line of input containing the decrypted text. The final letter 'A' should NOT be included. Punctuation, spaces, digits and any other non-letter characters are output unchanged.

### Sample input

```
N hfs wjfi ymnx xjsyjshj, xt rd fqltwnymr btwpx.F
Serq Unzvygba-Wbarf vf 85 gbqnl!  Unccl oveguqnl, Serq.N
Ocz lpdxf, wmjri ajs ephkzy jqzm ocz gvut yjbn.V
#
```

### Sample output

```
I can read this sentence, so my algorithm works.
Fred Hamilton-Jones is 85 today!  Happy birthday, Fred.
The quick, brown fox jumped over the lazy dogs.
```

# New Zealand Programming Contest 2007

## Problem I Recycling 30 Points

Kerbside recycling has come to New Zealand, and every city from Auckland to Invercargill has leapt on to the band wagon. The bins come in 5 different colours — red, orange, yellow, green and blue — and 5 wastes have been identified for recycling — Plastic, Glass, Aluminium, Steel, and Newspaper. Obviously there has been no coordination between cities, so each city has allocated wastes to bins in an arbitrary fashion. Now that the government has solved the minor problems of today (such as reorganising Health, Welfare and Education), they are looking around for further challenges. The Minister for Environmental Doodads wishes to introduce the "Regularisation of Allocation of Solid Waste to Bin Colour Bill" to Parliament, but in order to do so needs to determine an allocation of her own. Being a firm believer in democracy (well some of the time anyway), she surveys all the cities that are using this recycling method. From this she wishes to determine which allocation scheme (if imposed on the rest of the country) would cause the least impact, i.e. would cause the smallest overall number of changes to the current allocations.

Write a program that will read in a series of allocations of wastes to bins and then determine which allocation scheme, out of all possible schemes, will produce the fewest changes if adopted across the country. Note that there will always be a clear winner.

Input will consist of a series of blocks. Each block will consist of a series of lines terminated by a line containing only a single #, and each line will contain a series of allocations in the form shown in the example. The entire file will also be terminated by a line consisting of a single #.

Output will consist of a series of lines, one for each block in the input. Each line will consist of the allocation scheme that should be adopted as a national example. Bins should be listed in the order shown, i.e. red, orange, yellow, green, blue.

### Sample input

```
r/P,o/G,y/S,g/A,b/N
r/G,o/P,y/S,g/A,b/N
r/P,y/S,o/G,g/N,b/A
r/P,o/S,y/A,g/G,b/N
#
r/G,o/P,y/S,g/A,b/N
r/P,y/S,o/G,g/N,b/A
r/P,o/S,y/A,g/G,b/N
r/P,o/G,y/S,g/A,b/N
#
#
```

### Sample output

```
r/P,o/G,y/S,g/A,b/N
r/P,o/G,y/S,g/A,b/N
```

# New Zealand Programming Contest 2007

## Problem J Reversing Words 30 Points

“Oh no, not another reversing problem! We did those in Programming 101, years ago”, I hear you cry. Well yes, but this problem is just a little different, in that you have to reverse individual words, not entire sentences. Furthermore, only the letters are to be reversed — spaces, punctuation and case are unaffected.

For this problem a word is defined to be a sequence of characters containing at least two letters and delimited by whitespace. The letters in the word must be written in reverse order (so 'the' becomes 'eht'), but the other characters in the word remain in their original sequence. Furthermore, if a character in the original word is upper-case, the letter in the same position in the new word must also be uppercase, thus 'Smith-Jones' becomes 'Senoj-Htims'. The position of the word in the line must be unaltered, and the position of every non-letter must also be unaltered (so double blanks remain as double blanks).

Input will be a series of lines, terminated by a line consisting of a single '#'. No line will have more than 80 characters.

Output will consist of a series of lines, one for each line in the input, with the letters in each word reversed as described above.

### Sample input

```
This iS ;ate:'st  "";;   aBc.
Smith-Jones
#
```

### Sample output

```
Siht sI ;tse:'ta  "";;   cBa.
Senoj-Htims
```

# New Zealand Programming Contest 2007

## Problem K Word Crosses 30 Points

A word cross is formed by printing a pair of words, the first horizontally and the second vertically, so that they share a common letter. A leading word cross is one where the common letter is as near as possible to the beginning of both words, i.e. the sum of the positions of the common letter is minimised. If there are several minimal sums, choose the one that minimises the position of the matching letter in the first word of the pair. Thus MATCHES and CHEESECAKE could cross on 'A' (2+8 = 10), 'C' (4 + 1 = 5), 'H' (5+2 = 7), 'E' (7 + 3 = 10) and 'S' (7 + 5 = 12). The minimum is 5, so we use the 'C'. Double leading word crosses use two pairs of words arranged so that the two horizontal words are on the same line and each pair forms a leading word cross.

Write a program that will read in sets of four words (two pairs) and form them (if possible) into double leading word crosses, i.e. crossing the words in each pair separately, as in the example input below.

Input will consist of a series of lines, each line containing four words (two pairs) and terminated by a line consisting of a single #. In this problem, a word consists of 1 to 10 upper case letters.

Output will consist of a series of double leading word crosses as defined above. Leave exactly three spaces between the horizontal words. If it is not possible to form both crosses, write the message 'Unable to make two crosses'. Any trailing blanks will be stripped before judging. Leave one blank line between output sets.

### Sample input

```
MATCHES CHEESECAKE PICNIC EXCUSES
PEANUT BANANA VACUUM      GREEDY
A  VANISHING   LETTER TRICK
#
```

### Sample output

```
            E
            X
MATCHES   PICNIC
   H        U
   E        S
   E        E
   S        S
   E
   C
   A
   K
   E

Unable to make two crosses

V
A   LETTER
N     R
I     I
S     C
H     K
I
N
G
```

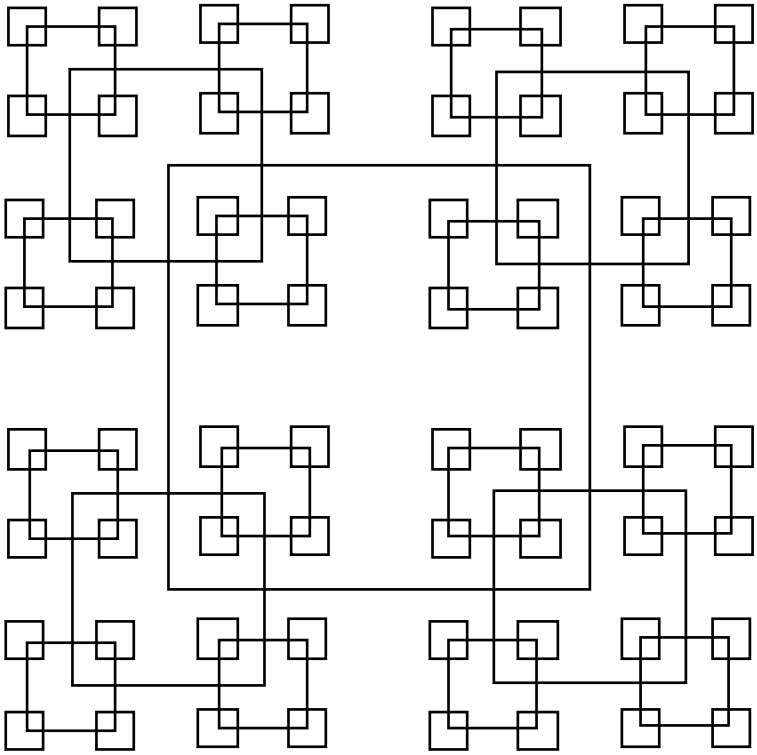
# New Zealand Programming Contest 2007

## Problem L All Squares 30 Points

Geometrically, any square has a unique, well-defined centre point. On a grid this is only true if the sides of the square span an odd number of points. Since any odd number can be written in the form $2k+1$, we can characterise any such square by specifying $k$, i.e. we can say that a square whose sides are of length $2k+1$ has size $k$. Now define a pattern of squares as follows.

1. The largest square is of size $k$ (i.e. sides are of length $2k+1$) and is centred in a grid of size 1024 (i.e. the grid sides are of length 2049).
2. The smallest permissible square is of size 1 and the largest is of size 512, thus $1 \le k \le 512$.
3. All squares of size $k > 1$ have a square of size $k / 2$ centred on each of their 4 corners. (Integer division, thus $9 / 2 = 4$).
4. The top left corner of the grid has coordinates (0,0).

Hence, given a value of $k$, we can draw a unique pattern of squares according to the above rules, e.g., if $k$ is 15, then the following pattern would be produced.



Obviously, any point in the grid will be surrounded by zero or more squares. (If the point is on the border of a square, it is considered to be surrounded by that square).

Write a program that will read in a value of $k$ and the coordinates of a point, and will determine how many squares surround the point.

Input will consist of a series of lines containing 3 integers ($k$ and the coordinates of the point) terminated by a line consisting of three zeroes (0 0 0).

Output will consist of a series of lines, one for each line of the input. Each line will consist of the number of squares surrounding the specified point.

### Sample input

```
500 113 941
0 0 0
```

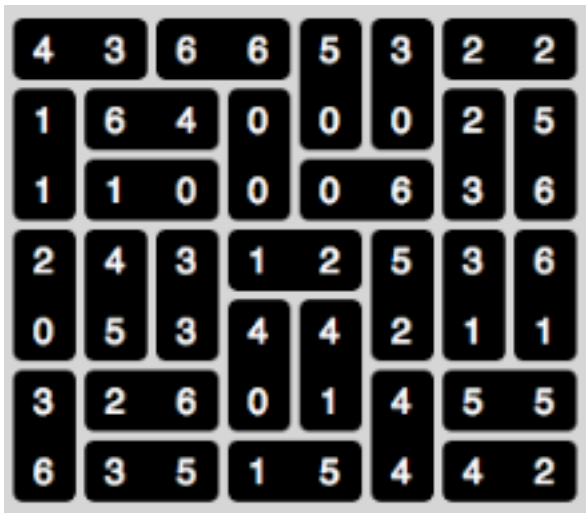
### Sample output

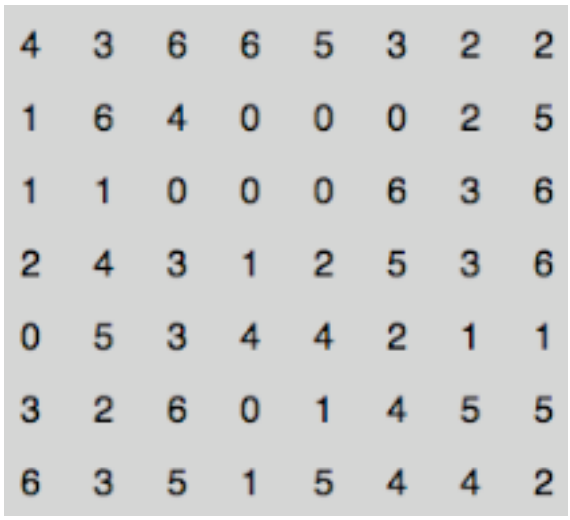
```
5
```

# New Zealand Programming Contest 2007

## Problem M Dominosa 100 Points

A set of dominoes, that is, one instance of every unordered pair of the numbers from 0 to $n$, has been arranged irregularly into a rectangle and the pattern has been recorded by writing the number in each square. For instance, in the figure below the original layout, using a 'standard' set ($n = 6$) of dominoes, is on the left and the puzzle configuration is on the right. Write a program to recreate the original configuration, guaranteed to exist and to be unique.




Input will consist of a series of puzzles. Each puzzle will start with a line containing an integer $n$ ($2 \le n \le 12$). Following this will be $n$ rows each containing $n$+1 characters from the first $n$+1 letters of the lower case alphabet, separated by single spaces. The set of puzzles will be terminated by a line containing a single zero (0). The above puzzle is represented in this form in the sample input.

Output will consist of a representation of the original grid in the same format as the input, except that a pair of letters constituting a horizontal domino are separated by an equals sign ('='), as shown in the sample output below. Leave a blank line between successive grids.

### Sample input

```
7
e d g g f d c c
b g e a a a c f
b b a a a g d g
c e d b c f d g
a f d e e c b b
d c g a b e f f
g d f b f e e c
0
```

### Sample output

```
e=d g=g f d c=c
b g=e a a a c f
b b=a a a=g d g
c e d b=c f d g
a f d e e c b b
d c=g a b e f=f
g d=f b=f e e=c
```

# New Zealand Programming Contest 2007

## Problem N Digit Sums 100 Points

Given 3 positive integers A, B and C, find how many positive integers less than or equal to A, when expressed in base B, have digits which sum to C.

Input will consist of a series of lines, each containing three integers, A, B and C, $2 \le B \le 100$, $1 \le A\ C \le 1{,}000{,}000{,}000$. The numbers A, B and C are given in base 10 and are separated by one or more blanks. The input is terminated by a line containing three zeros.

Output will be the number of numbers, for each input line (it must be given in base 10).

## Sample input

```
100 10 9
100 10 1
750000 2 2
1000000000 10 40
100000000 100 200
0 0 0
```

### Sample output

```
10
3
189
45433800
666303
```

# New Zealand Programming Contest 2007

## Problem O Stamps 100 Points

In far off Azeland they have a problem with their stamps. Inflation has been fierce, and the old denominations of 1, 5, and 12 lars now mean that for most letters there is not enough room to write the address after the stamps have been put on, so a government commission has been established to choose a new set of denominations.

They plan to keep the 1 lar stamp for historical reasons, but have discovered an obscure requirement in the postal legislation that requires it to be possible to make all "sufficiently large" amounts of postage without using the 1 lar stamp.

A tame mathematician at the university of Ogato has reassured them that this requirement will be met provided that the denominations of the remaining stamps have no common factor larger than 1. He did however mention that there is no known simple formula in that case to determine just what "sufficiently large" means precisely.

In order to evaluate the competing suggestions they would like to know first of all whether a set of denominations is acceptable at all and, if it is, what is the largest amount of postage that actually requires the use of the 1 lar stamp.

For instance, denominations of 9, 12, and 15 would be unacceptable, because they are all multiples of 3. Denominations of 4 and 7 would be acceptable, and the largest postage requiring the 1 lar stamp is 17 lar.

Input will consist of a number of sets of suggested denominations (the 1 lar stamp is omitted). Each denomination will be between 2 and 10,000 lar inclusive, and there will not be more than 10 denominations in a set. For each such set you must answer the questions above. Input is terminated by a sequence that begins with 0.

For each sequence, output consists of either the word "Unacceptable" or a single positive integer specifying the largest value not achievable with stamps of those values, i.e. the largest value that requires the use of a 1 lar stamp.

### Sample Input

```
9 12 15
4 7
3 5 7
0
```

### Sample Output

```
Unacceptable
17
4
```

# New Zealand Programming Contest 2007

## Problem P Assimilation 100 Points

Some of Farmer Brown's paddocks are overrun with rabbits, while the others contain sheep. Moreover, two malicious GEnies are playing a game. One, GEnie R, takes a paddock of sheep and converts them into rabbits. Not only that, but all the sheep in adjacent paddocks, and all the sheep in paddocks adjacent to those and so on are converted to rabbits at the same time. The other, GEnie S, does the same thing but converts a paddock of rabbits, its neighbouring rabbit paddocks, etc. to sheep. They do this alternately, and one of them wins when all the animals are of its type.

Farmer Brown is tired of not knowing when he wakes up in the morning whether he'll need his shotgun or his shears, so he'd like to be able to determine which of the GEnies is going to win (GEnies are very clever and you can be certain that if either one of them has a winning strategy then it will be used). The problem is to determine the winner (there always is one), given the name of the GEnie who will move first, together with the initial distribution of rabbits and sheep. For example, if GEnie R is playing first and the initial distribution is:

```
RSR
RSS
SRR
```

then he could move either of the following positions:

```
RRR     RSR
RRR     RSS
SRR     RRR
```

The first move would let GEnie S win on the next turn (all the rabbits form one big group so would become sheep *en masse*). But, in the other case, whether GEnie S makes all the lower left Rs into Ss, or the one in the upper right, it leaves just a single group of S's, and so GEnie R will win. So, in this case, GEnie R has a winning strategy.

Input will consists of a sequence of problems to solve, terminated by a line containing 0 0 #. Each problem will begin with a line containing two integers $r$ and $c$ ($1 \le r\ c \le 5$) denoting the size of the farm, where $r$ is the number of rows of paddocks and $c$ is the number of columns; and a single upper case letter ('R' or 'S') denoting which GEnie plays first. Such lines will be followed by $r$ lines consisting of $c$ upper case letters either 'R' or 'S' denoting rabbits or sheep respectively.

Output will consist of a single line for each problem in the input, containing only the character R or S according to whether GEnie R or GEnie S has a winning strategy.

### Sample input:

```
1 3 S
RSR
3 3 R
RSR
RSS
SRR
0 0 #
```

### Sample output

```
R
R
```