



ECOO 2006

Programming Contest

Board wide Contest

**to be written
after March 19th**

Problem 1: Fibonacci's Rabbits

The well known Fibonacci sequence 1,1,2,3,5,8,13,21,... is based on the proposition that, given that any pair of rabbits will produce a new pair of rabbits every month for ever, except in the first month after their birth. The sequence gives the number of pairs of rabbits that there would be in any given month.

That is:

- starting with 1 newborn pair in month one,
- which would still be 1 pair of rabbits in month two,
- which would be 2 pairs in month 3 (since this pair would produce a new pair in month 3),
- there would be 3 pairs in month 4 (since the first pair would produce a new pair)
- there would be 5 pairs in month 5 (since the first two pairs would produce a new pair... etc.)

But suppose, that, instead of living forever, and reproducing endlessly, each pair of rabbits would only live for 5 months, and therefore only produce 3 new pairs before they died. The sequence then would run: 1,1,2,3,5,6,10,14,...

And if each pair would live for only 3 months, the sequence would become: 1,1,2,1,2,1,2,...

DATA11.txt (DATA12 for the second try) contains 5 lines of two positive integers, A and B. The integer A represents the number of months each pair of rabbits lives, $2 < A < 101$. The integer B represents the number of months that have passed when the count of the number of pairs of rabbits occurs. A and B will have values such that the number of pairs of rabbits will never exceed 999 999 999.

Output should be on a cleared screen or window, and will state the number of rabbit pairs in each of the 5 cases. Output should appear as in the example below.

Sample Input:

```
5 10
30 30
12 40
4 40
6 49
```

Sample Output

```
in problem #1 there are 30 rabbit pair(s)
in problem #2 there are 832040 rabbit pair(s)
in problem #3 there are 95539192 rabbit pair(s)
in problem #4 there are 65657 rabbit pair(s)
in problem #5 there are 740444619 rabbit pair(s)
```

Problem 2: Flip Twice Cypher

Consider the 27 characters composed of a blank and the 26 capital letters of the alphabet:

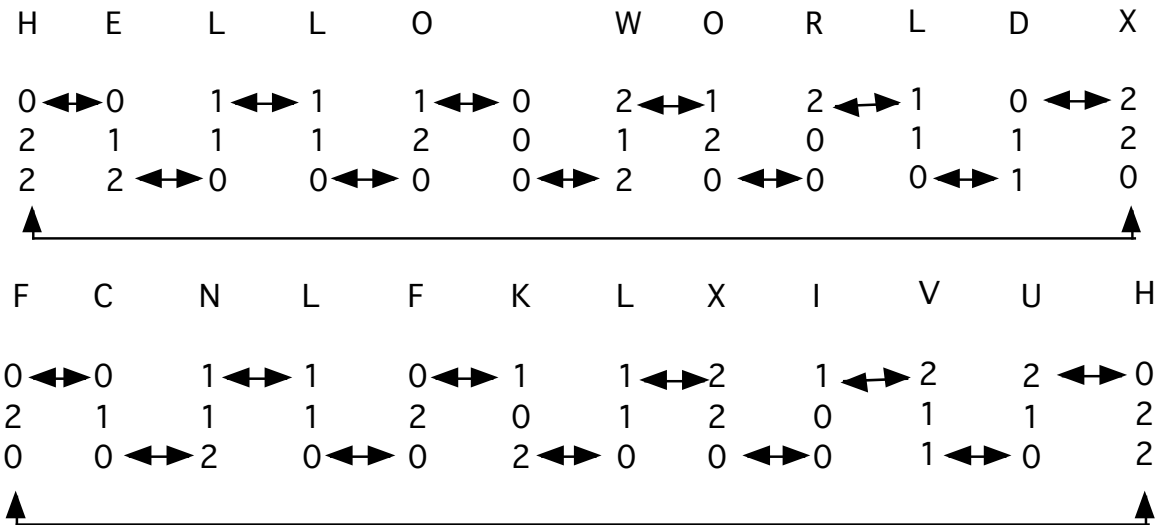
“ ABCDEFGHIJKLMNOPQRSTUVWXYZ”

They can be written, according to their position in the string, as a number between 0 and 26 inclusive. In base 3 these numbers would be written with exactly 3 digits, from 000 for blank to 222 for the last letter, “Z”.

The “Flip Twice Cypher” divides a message into pairs of characters as shown in the diagram below. In the example “HELLO WORLD X” there is an odd number of characters, so a dummy character (the X) is added to make a final pair. The most significant digit from the base 3 number of each letter is switched between each pair. The characters then form new pairs, by turning to the other neighbour as shown and switching the least significant digits. Since the first and last characters have no partner, they perform the switch with each other.

What results is a new set of base 3 numbers, which again can be turned into letters.

The coded message now reads: “FCNLFK LXIVUH”



DATA21.txt (DATA22 for the second try) contains 5 lines of text, each representing one encoded message. It is your job to decode this message and display it on the screen as in the sample below. No line will contain more than 250 characters. Output may wrap if it does not fit on one line. Note that if a line contains an odd number of characters, you must assume the existence of a last (phantom) blank character.

Sample Input: (only 3 of 5 lines are shown)

```
FCNLFK LXIVUH
LQCR RISBHCT KND OF SICCKIX CONDBIXRBHCB ALJFFB ZEAFISSZZ
BZEAIXJTCOJFFBRUHK XZM TDLABDLS SOIA TFCBW
```

Sample Output: (only 3 of 5 lines are shown)

```
HELLO WORLDX
NOW IS THE TIME FOR ALL TO COME TO THE AID OF THE PARTYX
THE SPRING OF TWO THOUSAND AND SIX IS HERE
```

Problem 3: Three Letter Words

A 3-letter word is a word composed of one central vowel surrounded by two consonants, or one central consonant surrounded by two vowels.

The vowels are: "AEIOU" and the consonants are: "BCDFGHJKLMNPQRSTVWXZ"

The letter Y is sometimes considered a vowel, and sometimes a consonant.

If the letter Y is used in one word, it is either used as a vowel only or a consonant only: **YYY** is not a valid word, but **HYT**, **ROY** and **EYE** are all valid.

DATA31.txt (DATA32 for the second try) contains 5 sets of data. Each data set contains two integers, A and B ($3 \leq A, B \leq 100$) on separate lines, followed by A lines of B letters each. It is your task to find the number of 3-letter words in the A x B rectangle. The words may be arranged in a total of four ways: horizontally, from left to right; vertically from up to down; diagonally from upper left to lower right; and diagonally from upper right to lower left.

Sample Input: (only 3 of 5 sets)

```
4
3
ASK
SLA
JFN
SNM
11
11
WUKFYTXIZDF
MNHBUAYDGA
NGUWOCJICQT
IKJJZTMXGZL
FOVGSKEFXEM
BOTKPUCTMRN
IADREUKHSDH
DUWBBMRSKOG
TKIZOWLAYYL
BJHGOVZDPG
CDTPASRTMHS
5
6
ZCZOYQ
XZGQOF
RIACXZ
AFVVIF
ALXWSG
```

Sample Output: (only 3 of 5 lines)

```
There are 1 word(s) in the 4 by 3 array
There are 75 word(s) in the 11 by 11 array
There are 17 word(s) in the 5 by 6 array
```

Note: the 17 words in the last array are:

From left to right: _____ ZOY, QOF, VIF

From top to bottom: _____ YOX, ZIF, GAV, OXI, XIS

Diagonally from upper left: _____ XIV, ZAV, CIG

Diagonally from upper right: _____ OGI, YQA, QOC, QAF, AFA, ZIW

Problem 4: Two Robots

Two space robots are initially located at position (0,0,0) in a Cartesian grid.

Robot A faces along the positive x-axis, robot B faces the negative x-axis. They are back to back, their feet in the direction of the negative y-axis, their heads in the direction of the positive y-axis. The positive z-axis is along the right side of robot A (and the left side of robot B).

The robots can turn left or right 90° and tip forward or backward 90°.

The robots obey commands of the form RTx, where R stands for either A or B (command for robot A or B), followed by T (which is a letter that stands for direction: L for left, R for right, U for up and D for down), and finally a non-negative integer, x where $0 \leq x \leq 999$.

The string of instructions "AL45AU190BR122AD8"

means:

robot A turns left 90° and moves a distance of 45 units in the direction it faces.

It is now at location (0, 0, -45).

robot A turns up (tips backwards 90°) and moves 190 units in the direction it faces.

It is now at location (0, 190, -45).

robot B turns right 90° and moves 122 units in the direction it faces.

It is now at (0, 0, -122).

robot A turns down (tips forward 90° and moves 8 units in the direction it faces.

It is now at (0, 190, -53)

To the nearest integer, the two robots are 202 units apart at the end of the set of instructions.

DATA31.txt (DATA32 for the second try) contains 5 strings of data. Each string contains at least one instruction and at the most 50 instructions. In each of the five cases, find the final position of each robot, and the distance between them to the nearest integer, as in the sample output below.

Sample input:

```
AL45AU190BR122AD8
AD64AD55BL102BL143BD145AL109BD144AL158AL96AD116AL37
BD141BR58AD64BR165BD185AR99
AL99BR111AR21AR122AU47AR205BD24BR79AU157
BD198AD170BD88BU20AD107AU108BR65BU167BL20AR105AR180
```

Sample output:

A is located at (0 , 190 , -53)

B is located at (0 , 0 , -122)

A and B are 202 units apart

A is located at (66 , 52 , -13)

B is located at (-1 , -145 , 102)

A and B are 238 units apart

A is located at (0 , -64 , 99)

B is located at (185 , 24 , -58)

A and B are 258 units apart

A is located at (-184 , 47 , -134)

B is located at (79 , -24 , -111)

A and B are 273 units apart

A is located at (-107 , -98 , 105)

B is located at (-79 , -238 , -65)

A and B are 222 units apart



ECOO 2006

Programming Contest

Regional Contest

Saturday, April 29 2006

Problem 1 – Split Code

The Spit-Code algorithm is a secret code where each character's ASCII code is first represented by two hexadecimal digits. The following steps are then performed on these digits:

- 1 Starting with the second digit, add the previous digit; if the result is a 2-digit hex number, ignore the high order digit;
- 2 Randomly replace the digit by the corresponding letter in the set: "ABCDEFGHJKLMNPO" or the set "QRSTUVWXYZKLMNOP"
Where 0 is replaced randomly by either "A" or "Q", 1 is replaced by "B" or "R" etc.

For example: Our school is the best
in hex code: 4F7572207363686F6F6C206973207468652062657374

Then, since $4+F = 13$, the F is replace by 3 (since the high order digit, 1, is ignored)
And since $3+7 = A$, the 7 is replaced by A

And so on: 43AF68AA14AD3B1065B799F8F244BF5D38AA028D47E2
Finally: ETKPWIKKBEKNTLBAGFLXZZPIPSUULPFNTIKKQOSINUXOS

The use of the letters above P is important, so that potential codebreakers do not realize that the code is based on the number 16.

DATA11.txt (DATA12.txt for the second try) contains 5 lines, each of which represent a message which is independently encoded using the Split-Code algorithm. It is your task to write a program that will decode these 5 lines. Each line contains 250 characters or less.

Output should be on a cleared screen or window. Since some messages will wrap and be displayed on more than one line, **separate messages with a blank line**.

Sample Input:

```
ECYXOVHHNPGPRRYMCKAZATVVMAGPVCINPPUXNOESILRJPOQAWOUVLP RRYKQZATJOUCEA
HZPQHLBZPELNPPWKAYODFVMTZIPRYLBWYIOEKZASUULPFNTYKKQYOPVDJNDPFOUCYP
GFLBDDKOU MCHZJPVMOENTULNEHJQCCYVMBXTJOFZPEKPGIPC
GXNLRFHXHORYMCRHGMQCSJQGHOSYLRZPYOMCZ
HLBZPUWWMOUFMQXLBN TIKKAPVLNNTLRKREWWMZPQHKBFLAHZLXZZNBXGMKMMRSJOUNEMCRYMSH
```

Sample Output

```
Now by this time Sancho had risen,
rather the worse for the handling
of the friars' muleteers
and stood watching
the battle of his master, Don Quixote
```

Problem 2 – Sum of Squares

Any nonnegative integer can be written as the sum of at most 4 squares:

$$1 = 1^2$$

$$5 = 2^2 + 1^2$$

$$9999 = 99^2 + 14^2 + 1^2 + 1^2$$

Interestingly, it seems that only numbers that have a remainder of 7 when divided by 8, are written as a sum of 4 squares. All others can be expressed as the sum of 3 squares or less.

Write a program that can take a number and express it as the sum of one or more squares. You must use the minimum number of squares possible. For example, 4 can be expressed as $1^2+1^2+1^2+1^2$, however, 4 should be expressed as 2^2 .

You must list the squares in descending order: the largest squares first, the smallest squares last.

In many cases there are more than one solution. You must choose the solution where the first number is as large as possible, and if there is more than one such choice, pick the one where the next number is as large as possible.

For example, there are many solutions for the case of 9999. There are however only four solutions containing 99^2 as one of its terms:

$$9999 = 99^2 + 9^2 + 9^2 + 6^2$$

$$9999 = 99^2 + 10^2 + 7^2 + 7^2$$

$$9999 = 99^2 + 13^2 + 5^2 + 2^2$$

$$9999 = 99^2 + 14^2 + 1^2 + 1^2$$

And in this case, the second term, 14^2 , indicates the proper choice.

DATA21.txt (DATA22.txt for the second try) contains 5 numbers between 1 and 9999 (inclusive). Express each number as a sum of squares, as detailed above.

Sample Input:

```
4
1234
23
456
9999
```

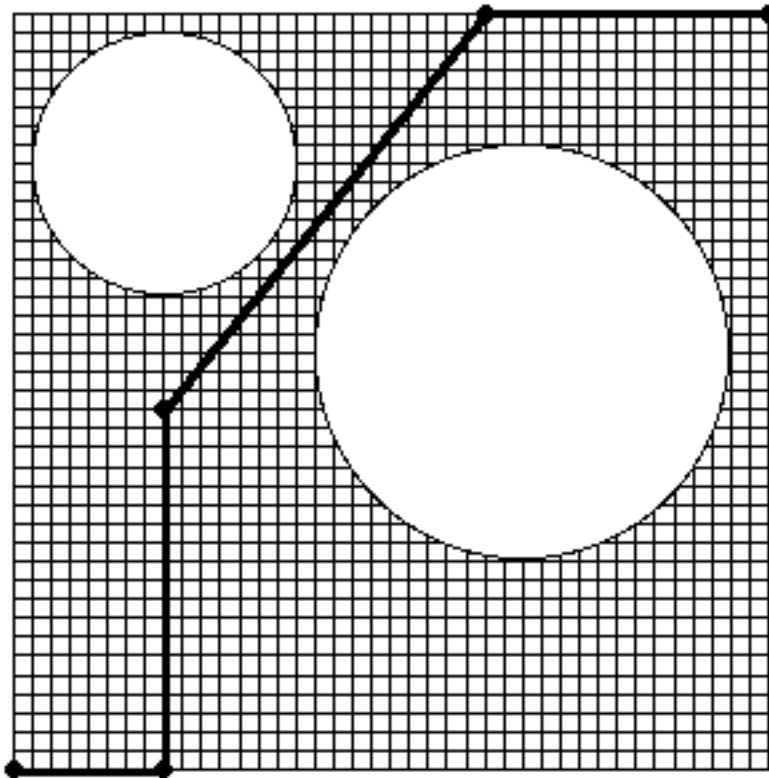
Sample Output

```
4 = 2 squared
1234 = 35 squared + 3 squared
23 = 3 squared + 3 squared + 2 squared + 1 squared
456 = 16 squared + 14 squared + 2 squared
9999 = 99 squared + 14 squared + 1 squared + 1 squared
```


Problem 3 – Polynomial Path

Consider a 40x40 grid, where the lower left corner is (0,0) and the lower right corner is (40, 0) and the upper right (40,40). Within this grid lie two circles, with integer centers, and integer radii. These circles are separated from each other by at least 2 units and neither of them touch any of the four sides of the grid.

Write a program that draws a polynomial path from (0,0) to (40,40) that passes between the two circles. The path may touch the edges of the grid but may not touch the two circles.



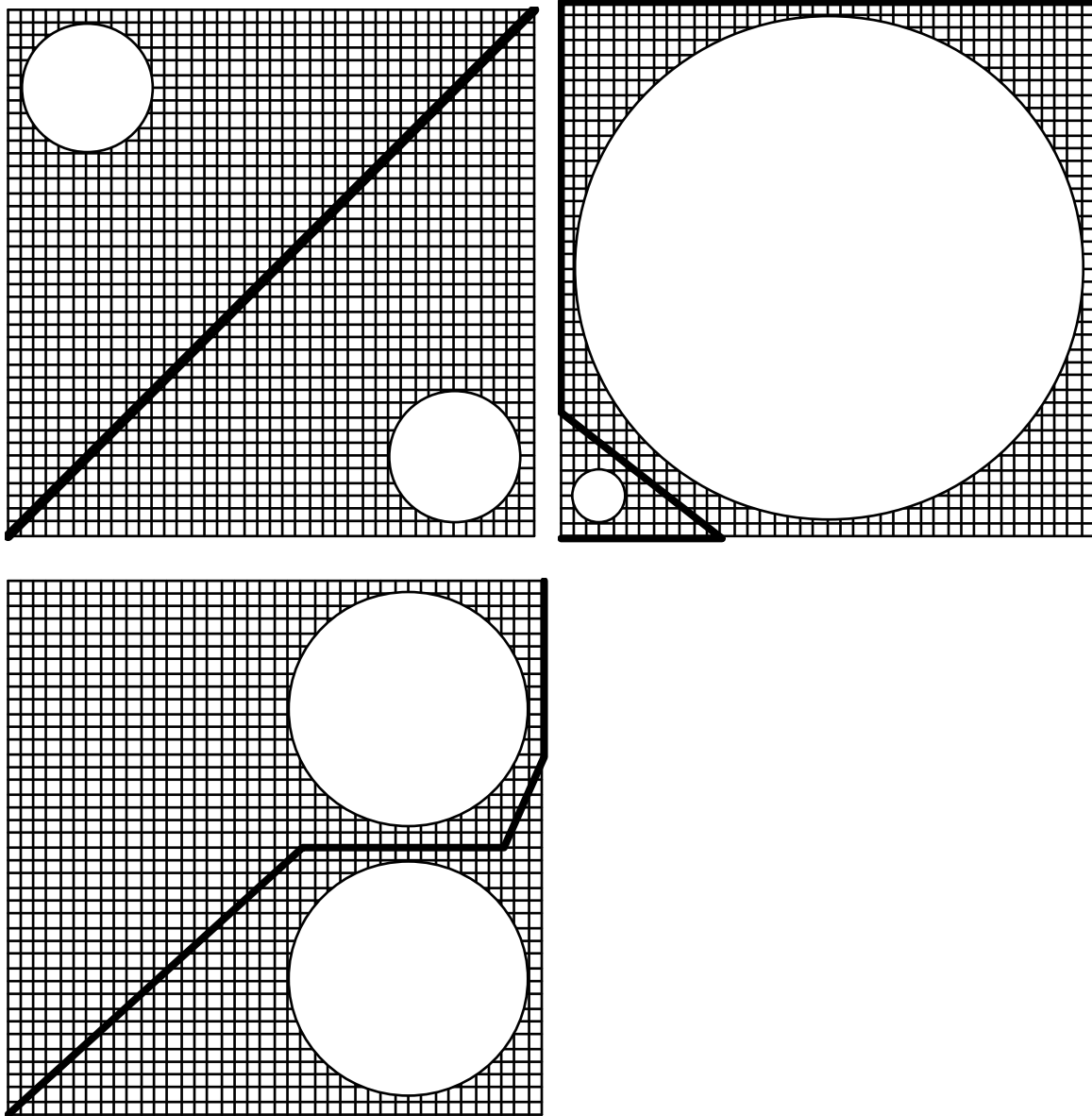
In this example two circles are given, the first one has the center at (8,31) with radius 7, the second circle has its center at (27,22) with radius 11. The path shown starts at (0,0), passes through (8,0), (8,19) and (25,40) and ends at (40,40).

DATA31.txt contains ten lines, representing five sets of two lines. Each line contains 3 positive integers representing the x-coordinate, the y-coordinate and the radius of a circle. Your task is to draw 5 consecutive pictures, each on a separate window or screen, with a prompt, to continue with a keystroke or a mouse click. Each picture must contain a 40x40 grid, the two circles and a path from (0,0) to (40,40) as described.

Sample input (only 3 of 5 examples are shown)

6 34 5
34 6 5
20 20 19
3 3 2
30 30 9
30 10 9

Sample output



Problem 4 – Cryptarithm

A cryptarithm is a mathematical expression in words where each letter represents a digit. The most famous of these is:

SEND		9567
<u>+MORE</u>	Which is uniquely	<u>+1085</u>
MONEY	represented by the sum:	10652

That is: O=0, M=1, Y=2, E=5, N=6, D=7, R=8 and S=9

Write a program that will solve sums of this kind. You may assume for the purpose of this program, that all sums are composed of 3 words, and that the third word (the sum) has one more letter than any of the two addends and, by implication, the first letter of that word represents a 1. Note also, that the first letter of any word cannot represent a 0.

DATA41.txt (DATA42.txt for the second try) contains 5 sets of 3 words on three lines, for a total of 15 lines. The first two words of the set are the addends; the last word of the set of three is the sum. Your output should be formatted as shown below.

The program may consume a lot of time. **Note that only the solutions that are correctly displayed within the first minute of the program will be counted for points.**

Sample Input:

```
AS
A
MOM
DOWN
WWW
ERROR
OOOH
FOOD
FIGHT
TAKE
THAT
SHEET
LEAH
LOVES
RUSSIA
```

Sample Output

```
AS      92
A       9
MOM     101
DOWN    9364
WWW     666
ERROR   10030
OOOH    8886
FOOD    1883
FIGHT   10769
TAKE    7460
THAT    7547
SHEET   15007

LEAH    9325
LOVES   98437
RUSSIA  107762
```



ECOO 2006

Programming Contest

Final Contest

Saturday, May 13 2006

Problem 1 – Fuel Crisis

Salsa, a fruit salesman, living in Appletown, has his usual route through Banana city, Cabbage town, Dill place, Endive junction, Figville and Grape corners (A,B,C,D,E,F and G for short). Unfortunately the donkey that pulls his cart only runs on carrots, and as long as he can feed it a carrot every kilometer, it will happily pull the cart at 10 kph. Unfortunately, it refuses to work without the carrot, so when our salesman runs out of carrots, he has to hitch the donkey to the back of the wagen and pull the load himself... at 1 kph.

Now the distances in kilometers between the villages is as indicated below:

	<i>Apple town</i>	<i>Banana city</i>	<i>Cabbage town</i>	<i>Dill place</i>	<i>Endive junction</i>	<i>Fig ville</i>	<i>Grape Corners</i>
Apple town	0	28	22	18	32	36	20
Banana city	28	0	17	39	52	40	23
Cabbage town	22	17	0	20	45	40	37
Dill place	18	39	20	0	28	38	46
Endive junction	32	52	45	28	0	22	47
Fig ville	36	40	40	38	22	0	30
Grape Corners	20	23	37	46	47	30	0

In each town of course, Salsa buys all the available supply of carrots, for he hates to pull the wagen himself. It is your task to find the route for Salsa to travel, where he spends the least amount of time on the road. Make sure Salsa is back in Apple town at the end of the trip.

DATA11.txt (DATA12.txt for the second try) contains 5 lines, each containing 7 integers. The 7 integers represents the number of carrots Salsa can buy in each town, in order, from Apple town to Grape Corners. Print out the order of the villages he must visit, and how many hours he spends on the road, to one digit accuracy.

Sample input

```
9 0 50 15 8 5 5
10 10 5 70 80 0 0
30 10 10 50 10 40 0
0 25 15 25 15 35 15
10 20 30 40 30 20 10
```

Sample output

```
The best route is: ACBGFEDA with total time = 77.2 hours
The best route is: ADEFGBCA with total time = 23.2 hours
The best route is: ADEFGBCA with total time = 25.0 hours
The best route is: ADEFGBCA with total time = 43.9 hours
The best route is: ADCBGFEA with total time = 23.4 hours
```

Problem 2 – Blobs

A blob is an arrangement of at least 5 letters of the same kind that are connected. Any two letters are considered connected, if they are neighbours left-right or neighbours up-down. Diagonal neighbours are not considered connected. In the rectangle below, for example, there is a Q-blob, a B-blob and a Y-blob. The group of A's near the upper left corner do not form a blob, because there are only 4 A's that are connected. (the A-group has two diagonal neighbours that are NOT connected.)

```
XAAFZQXBYYVED
PAAPXQQCLYQWZ
AMIAATQQACYFUT
BPSBBQZMKYHLJ
RVBBBBQSNYAPZ
GCIBBHKIZYFPY
```

Write a program that finds the number of bobs and the number of letters in the largest blob. If a particular rectangle is blob-free, indicate that with an appropriate statement, as in the example below.

DATA21.txt (DATA22.txt for the second try) contains 5 sets of data. Each set is composed of a line containing two numbers, x and y followed by x lines of y characters. Note that $4 < x, y < 50$.

Sample Input

```
6 10      13 13      8 7      5 5      7 7
AAXBFUJYAD  NAXUEEQWMBELG  XXBBRRH  SSSSS  ABCDEFG
AAWWANNJYT  NFCFVMTBGNEQG  XKBBRRH  SSSSS  ABCDEFG
KWWWWNNNNN  BXJLYNZXGVOBG  YKKKBHH  SSSSS  ABCDEFG
KKBMEGVCMK  DUWXVCVKNIMOG  BDKKHHH  SSSSS  ABCCCCC
KKPPGYYYQQ  HJEETCRMANUGG  MKQDUCI  SSSSS  ABCCDCC
KPPPPKYJQQ  AIEERCCCUOGL  QQONPXV  ABCCDDC
              JMHPYJLGGGGGZ  RFQLPPD  ABCCCCC
              RFHLVPDGYYYYY  RFHLVPP
              FVVFOMJGGGGGG
              VQVVVVVOSODWN
              KIHCMOONWQQQ
              IFXXXWOIWMQQS
              QPLXXOOOLWQTK
```

Sample Output

```
There are 4 blobs, the largest contains 7 letters
There are 7 blobs, the largest contains 19 letters
There are 4 blobs, the largest contains 7 letters
There are 1 blobs, the largest contains 25 letters
There are 3 blobs, the largest contains 19 letters
```

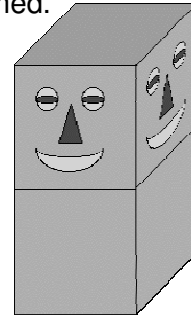
Problem 3 – Tumbledee

Tumbledee is a 2x1x1 block that can tumble in any one of four different directions: North, South, East or West. It can be standing Up or laying Down in any one of six different positions, depending on where its head is in relation to its feet: North, South, East, West (when it is laying down) and Up or Down (depending on when Tumbledee is standing upright or on its head).

Tumbledee stands in the middle of a board like a chess board, except, that the squares extend indefinitely in all four directions. Tumbledee's initial position is on square (0,0). The square immediately to its north is (0,1); immediately to its south is (0,-1), to its east is (1,0) and to its west is square (-1,0). All other squares are numbered consistently, and so for example the square 5 positions to its east and 4 positions to its north is named (5,4).

Tumbledee starts tumbling according to a string of instructions made up of the four letters E,W,N,S indicating the four directions, and the challenge is to keep track of where the head of Tumbledee is located after the series of tumbles it has performed.

For example, the series of tumbles WEEENSS would place its head
W — first at (-2,0), laying down with its head pointing West.
E — then back at (0,0), being again in the Up position,
E — followed by (2,0), pointing East
E — followed by (3,0), now being upside down
N — followed by (3,1), laying down again, pointing South
S — followed by (3,0), again standing on its head
S — followed by (3,-1) with its head pointing North.



DATA31.txt (DATA32.txt for the second try) contains 5 lines of data. Each line represents a set of tumbles and contains no more than 200 letters. Your task is to write a program that will be able to locate Tumbledee's head at the end of each set of tumbles, while assuming that Tumbledee starts out in square (0,0) and in the upright position at the start of each set.

Sample Input:

```
NENNEESWNN
ENEENNNWNNENENNNWSENEEESESWSSES
NEEESSENEEESENEEEESWWNEEESWWSSSSWSEENEEESWWN
WNESWSSWWNNNEEESWWS
E
```

Sample Output

```
Tumbledee's head is at (3,5)
Tumbledee's head is at (12,7)
Tumbledee's head is at (14,-7)
Tumbledee's head is at (-2,-1)
Tumbledee's head is at (2,0)
```

Problem 4 – Word Chain

A well known word game involves changing one word into another word by successively substituting **one letter at a time**, by another letter, in such a way that the intermediate words still make sense.

For example, to change the word ALONE into CRAZY could involve the following steps:

ALONE – CLONE – CRONE – CRANE – CRAZE - CRAZY

Each of these intermediate words are acceptable English words.

Sometimes you are forced to extend the chain to several more words, to ensure that each intermediate word be found in the dictionary.

In this problem we will only consider 5-letter words, and for convenience, there is a “dictionary” of 5-letter words in the input file.

DATA41.txt (DATA42.txt for the second try) contains 10 lines each containing one 5-letter word in capitals, representing 5 sets of two words. The first word is the starting word of the chain, the second the final word. These ten lines are followed by the “dictionary”, composed of an integer n (on line 11), where $30 < n < 50$, followed by n lines, each containing a 5 letter word (in capital letters and in alphabetical order).

Output should be the shortest chain possible, as shown in the sample below.

There will always be a solution.

Sample Input (37 lines)

ALONE	26	CRONE	GROWL
CRAZY	ALONE	CROSS	GROWS
CROSS	BLOBS	GLOBE	TRIAL
GLORY	BRAIL	GLOBS	TRISL
GROWL	BRAIN	GLOBS	TRIST
BLOBS	BROIL	GLORE	TRUSL
BRAIN	CLONE	GLORY	TRUST
BROIL	CRANE	GLOSS	
TRIST	CRAZE	GLOWS	
TRIAL	CRAZY	GROSS	

Sample Output

ALONE – CLONE – CRONE – CRANE – CRAZE – CRAZY
CROSS – GROSS – GLOSS – GLOBS – GLOBE – GLORE – GLORY
GROWL – GROWS – GLOWS – GLOBS – BLOBS
BRAIN – BRAIL – BROIL
TRIST – TRISL – TRIAL