NOIP 普及组复赛

冲刺赛

问题	Problem1	Problem2	Problem3	Problem4
题目	Do the Untwist	Fire Net	Gridland	Great Equipment
时间限制	2s	2s	2s	10s
内存限制	65536 KB	65536 KB	65536 KB	32768 KB

Problem 1

Cryptography deals with methods of secret communication that transform a message (the ***plaintext***) into a disguised form (the ***ciphertext***) so that no one seeing the ciphertext will be able to figure out the plaintext except the intended recipient. Transforming the plaintext to the ciphertext is ***encryption***; transforming the ciphertext to the plaintext is ***decryption***. ***Twisting*** is a simple encryption method that requires that the sender and recipient both agree on a secret key *k*, which is a positive integer.

The twisting method uses four arrays: *plaintext* and *ciphertext* are arrays of characters, and *plaincode* and *ciphercode* are arrays of integers. All arrays are of length *n*, where *n*is the length of the message to be encrypted. Arrays are origin zero, so the elements are numbered from 0 to *n* - 1. For this problem all messages will contain only lowercase letters, the period, and the underscore (representing a space).

The message to be encrypted is stored in *plaintext*. Given a key *k*, the encryption method works as follows. First convert the letters in *plaintext* to integer codes in *plaincode*according to the following rule: '_' = 0, 'a' = 1, 'b' = 2, ..., 'z' = 26, and '.' = 27. Next, convert each code in *plaincode* to an encrypted code in *ciphercode* according to the following formula: for all *i* from 0 to *n* - 1,

$$ciphercode[i] = (plaincode[ki \bmod n] - i) \bmod 28.$$

(Here *x* mod *y* is the positive remainder when *x* is divided by *y*. For example, 3 mod 7 = 3, 22 mod 8 = 6, and -1 mod 28 = 27. You can use the C '%' operator or Pascal 'mod' operator to compute this as long as you add *y* if the result is negative.) Finally, convert the codes in *ciphercode* back to letters in *ciphertext* according to the rule listed above. The final

twisted message is in *ciphertext*. Twisting the message cat using the key 5 yields the following:

Array	0	1	2
plaintext	'c'	'a'	't'
plaincode	3	1	20
ciphercode	3	19	27
ciphertext	'c'	's'	'.'

Your task is to write a program that can untwist messages, *i.e.*, convert the ciphertext back to the original plaintext given the key *k*. For example, given the key 5 and ciphertext 'cs.', your program must output the plaintext 'cat'.

The input file contains one or more test cases, followed by a line containing only the number 0 that signals the end of the file. Each test case is on a line by itself and consists of the key *k*, a space, and then a twisted message containing at least one and at most 70 characters. The key *k* will be a positive integer not greater than 300. For each test case, output the untwisted message on a line by itself.

Note: you can assume that untwisting a message always yields a unique result. (For those of you with some knowledge of basic number theory or abstract algebra, this will be the case provided that the greatest common divisor of the key *k* and length *n* is 1, which it will be for all test cases.)

Example input:

5 cs.

101 thqqxw.lui.qswer

3 b_ylxmhzjsys.virpbkr

0

Example output:

cat

this_is_a_secret

beware._dogs_barking

Problem 2

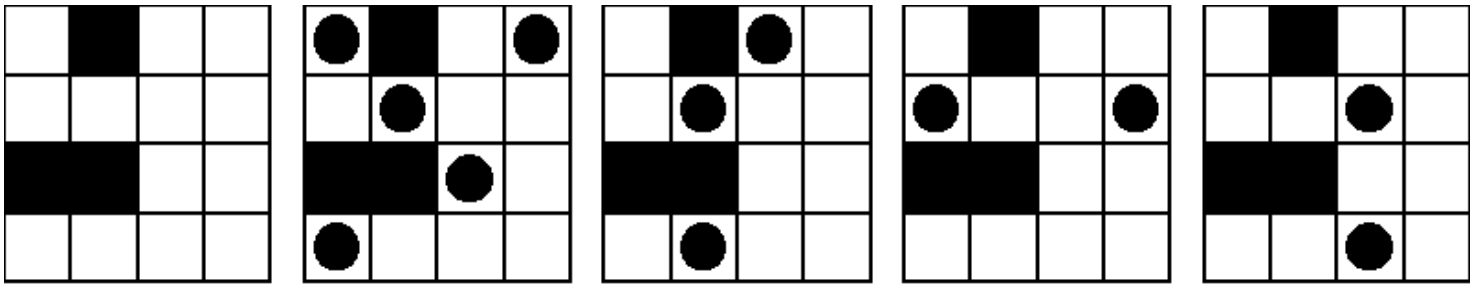
Suppose that we have a square city with straight streets. A map of a city is a square board with n rows and n columns, each representing a street or a piece of wall.

A blockhouse is a small castle that has four openings through which to shoot. The four openings are facing North, East, South, and West, respectively. There will be one machine gun shooting through each opening.

Here we assume that a bullet is so powerful that it can run across any distance and destroy a blockhouse on its way. On the other hand, a wall is so strongly built that can stop the bullets.

The goal is to place as *many* blockhouses in a city as possible so that no two can destroy each other. A configuration of blockhouses is ***legal*** provided that no two blockhouses are on the same horizontal row or vertical column in a map unless there is at least one wall separating them. In this problem we will consider small square cities (at most 4x4) that contain walls through which bullets cannot run through.

The following image shows five pictures of the same board. The first picture is the empty board, the second and third pictures show legal configurations, and the fourth and fifth pictures show illegal configurations. For this board, the maximum number of blockhouses in a legal configuration is 5; the second picture shows one way to do it, but there are several other ways.



Your task is to write a program that, given a description of a map, calculates the maximum number of blockhouses that can be placed in the city in a legal configuration.

The input file contains one or more map descriptions, followed by a line containing the number 0 that signals the end of the file. Each map description begins with a line containing a positive integer *n* that is the size of the city; *n* will be at most 4. The next *n* lines each describe one row of the map, with a '.' indicating an open space and an uppercase 'X' indicating a wall. There are no spaces in the input file.

For each test case, output one line containing the maximum number of blockhouses that can be placed in the city in a legal configuration.

Sample input:

4

.X..

....

XX..

....

2

XX

.X

3

.X.

X.X

.X.

3

...

.XX

.XX

4

....

....

....

....

0

Sample output:

5

1

5

2

4

Problem 3

Background

For years, computer scientists have been trying to find efficient solutions to different computing problems. For some of them efficient algorithms are already available, these are the "easy" problems like sorting, evaluating a polynomial or finding the shortest path in a graph. For the "hard" ones only exponential-time algorithms are known. The traveling-salesman problem belongs to this latter group. Given a set of N towns and roads between these towns, the problem is to compute the shortest path allowing a salesman to visit each of the towns once and only once and return to the starting point.

Problem

The president of Gridland has hired you to design a program that calculates the length of the shortest traveling-salesman tour for the towns in the country. In Gridland, there is one town at each of the points of a rectangular grid. Roads run from every town in the directions North, Northwest, West, Southwest, South, Southeast, East, and Northeast, provided that there is a neighbouring town in that direction. The distance between neighbouring towns in directions North-South or East-West is 1 unit. The length of the roads is measured by the Euclidean distance. For example, Figure 7 shows 2 * 3-Gridland, i.e., a rectangular grid of dimensions 2 by 3. In 2 * 3-Gridland, the shortest tour has length 6.

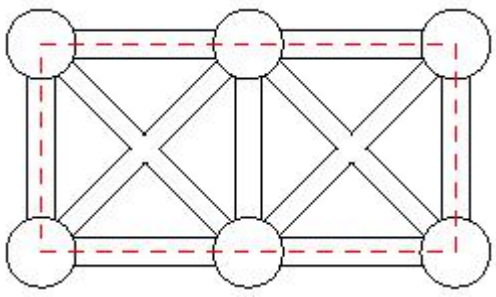


Figure 7: A traveling-salesman tour in 2 * 3-Gridland.

Input

The first line contains the number of scenarios.

For each scenario, the grid dimensions m and n will be given as two integer numbers in a single line, separated by a single blank, satisfying $1 < m < 50$ and $1 < n < 50$.

Output

The output for each scenario begins with a line containing "Scenario #i:", where i is the number of the scenario starting at 1. In the next line, print the length of the shortest traveling-salesman tour rounded to two decimal digits. The output for every scenario ends with a blank line.

Sample Input

2

2 2

2 3

Sample Output

Scenario #1:

4.00

Scenario #2:

6.00

Problem 4

Once upon a time, there lived Catherine Ironfist, the Queen of Enroth. One day, she received the news of her father's death. So she sailed for Erathia to attend her father's funeral. Fearing the worst, she assembled a military fleet as her escort. On reaching the coast of Erathia, Catherine found an allied wizard's tower, devastated from battle and abandoned. There she learned that a black-hearted knight poisoned her father using a goblet of wine, and Erathia was falling to the enemies. And then, she mustered local armies, and marched to Erathia's castle, restoring lost land along the way.

During the battles, she found that the equipments for the soldiers were in urgent need. And she knew clearly that the best equipments were made by the battle dwarf's workshop in the world. The workshop's equipments were well known for the firmness and top-quality. "Cloak of the Undead King", "Armor of the Damned", "Angelic Helm" are the nonesuch ones. But unfortunately, the workshop was seated at the Erathia's castle, the territory under the enemy's control. So she sent a brave volunteer to come into the castle and asked for the workshop's help.

"It's our pleasure to help the righteous heroine." Rion, the leader of the workshop sent the message to Catherine, " We haven't enough resources to build the nonesuch equipments. So we'll try to offer the ordinary equipments as more as possible. Still, those ones are much better the equipments made by other workshops. But we have faced a difficult problem. The castle is in a state of siege. The guards prohibited the equipments to be carried away from the castle. We have to ask for the trade caravans' help. As you know, each trade caravan's capability of carrying equipments is limited. If they had carried a little more, they would be

detected by the guards, which would lead them into the prison. So you have to arrange how to carry these equipments."

The workshop would offer helms, armors and boots. These three ones had different defend capabilities. Also, their weight and size were different from each other. What's more, Rion had told Catherine that if armed with those three ones together, they would form a set of equipments, which might provide much more defend capability. As far as the trade caravan was concerned, each one had its special weight limitation and size limitation. Catherine had to make the decision on how to arrange the transportation plan to provide her soldiers as more defend capabilities as possible. Could you help her to finish the plan?

Input

The input describes several test cases. The first line of input for each test case contains a single integer n, the number of trade caravans (0 <= n <= 100).

The following four lines describe the information of those equipments. The first line contains three integers w1, s1 and d1, indicating the weight, size and defend capabilities of the helm. The integers w2, s2 and d2 in the second line represent the weight, size and defend capabilities of the armor. Also, in the third line, w3, s3 and d3 are the weight, size and defend capabilities of the boot. The fourth line contains four integers c1, c2, c3 and d4. Among those integers, c1, c2, c3 are the number of helms, armors and boots in a set of equipments, d4 is the capability of this set.

In the test case, following those data are n lines, describing the carrying capabilities of the trade caravans. Each line contains two integers, xi and yi, indicating the weight limit and size limit of a trade caravan.

The input is terminated by a description starting with n = 0. This description should not be processed.

Note: Because of the trade caravans' carrying capabilities, you may assume the quantities of the helms, armors and boots will not exceed 500 respectively.

Output

Your program must compute the defend capability of the best carrying plan. That is, after having performed the carrying plan, the defend capability of the equipments which have been carried away from the castle should be the largest. For each test case in the input file, print the case number and a colon, and then the defend capability of those equipments.

Print a blank line between test cases.

Sample Input

```
3
1 1 3
5 6 10
```

2 1 2

1 1 1 50

1 1

5 6

2 1

0

Output for the Sample Input

Case 1: 50