Problem Set for the 2nd Day of Winter Camp

Japanese Alumni Group

Contest Held on 23 Feb 2008

Status of Problems

All problems were newly created by the members of Japanese Alumni Group.

Some portion of the problem statement may be modified for correction and/or clarification.

Terms of Use

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem A

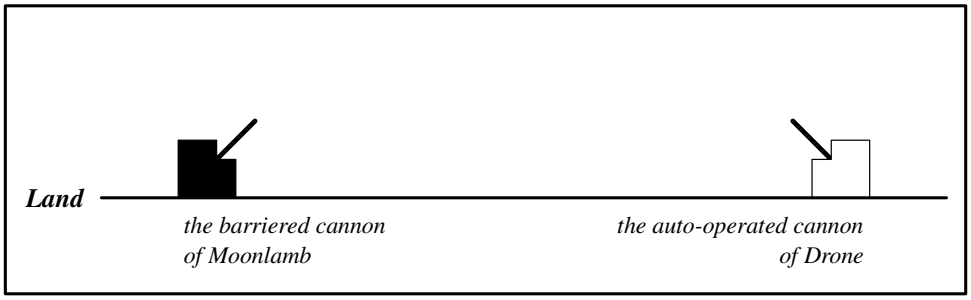
Defend the Nation

Input File: defend.in

Two nations Drone and Moonlamb have been engaged in a war. You have been called up for service as a talented programmer in Drone.

Recently, the army of Moonlamb has set up a new cannon on the field. The army of Drone knows exactly where it has been placed, but not how it can be destroyed; direct attacks would not be successful because the army of Moonlamb has also set up a special barrier around the cannon. Fortunately, it seems that no projectiles have reached the troops yet. Nevertheless, the army of Drone immediately requires an effective means to intercept projectiles fired from the cannon. Otherwise they would definitely take severe damage once the army of Moonlamb started attacks with the cannon.

The generals of Drone have decided to build a new interception system that consists of a radar system, a control program, and an auto-operated cannon. As neither of the cannons of Moonlamb and Drone can change the firing directions except the elevation angles, the auto-operated cannon will be placed so the two barrels face each other. This way the army of Drone will have a better possibility of successful interception. The figure below depicts this situation.



The interception system works as follows. When the radar system detects projectiles fired from the barriered cannon, it immediately sends their initial velocities and elevation angles to the control program. Then the control program determines the elevation angles at which the interceptors should be fired in order to destroy the targeted projectiles on the fly, and sends firing commands to the auto-operated cannon. Finally, the auto-operated cannon fires interceptors according to the received commands.

For the maximum safety of the auto-operated cannon and other installations behind, the army of Drone wants to destroy the projectiles at the earliest possible time. On the other hand, the interceptors can be fired only with their fixed initial velocities. The control program thus can only control the elevation angles and the firing times. In addition, the interceptors cannot be fired in less than τ seconds from when the targeted projectiles are fired, because of delay in detecting those projectiles.

The trajectories of the projectiles and the interceptors shape parabolas, both affected only by the gravity with the acceleration of 9.8 [m/s^2]. The sizes of the projectiles, the interceptors, and the cannons are all negligible.

Your task is to write the control program and defend the nation Drone!

Input

The first line of the input contains a single integer that indicates the number of test cases.

Each test case consists of a line with five integers d, τ, v_1, θ_1, and v_2, in this order. d denotes the distance between the two cannons in meters ($1 \leq d \leq 1000$); τ denotes the minimum time needed in seconds between the projectile and the interceptor ($0 \leq \tau \leq 100$); v_1 denotes the initial velocity of the projectile in meters per second ($1 \leq v_1 \leq 1000$); θ_1 denotes the elevation angle of the targeted projectile in degrees ($1 \leq \theta_1 \leq 89$); and v_2 denotes the initial velocity of the interceptor in meters per second ($1 \leq v_2 \leq 1000$).

Output

For each test case, your program should print the elevation angle of the interceptor, and the time elapsing between the projectile being fired from Moonlamb and it being destroyed. The angle and the time should be given in degrees and seconds respectively, and both should be printed with exactly six fractional digits. In case of multiple angles giving the same minimum elapsing time, your program may print any of them. If it is impossible to destroy the projectile before reaching the land, your program should report as such.

The output should be formatted as shown in the sample below. Print an empty line between test cases.

Sample Input	Output for the Sample Input
2 150 1 50 30 50 100 1 20 30 40	Case 1: Fire at 37.995392 degree(s). Projectile destroyed in 2.290123 second(s). Case 2: Projectile reaches the land.

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

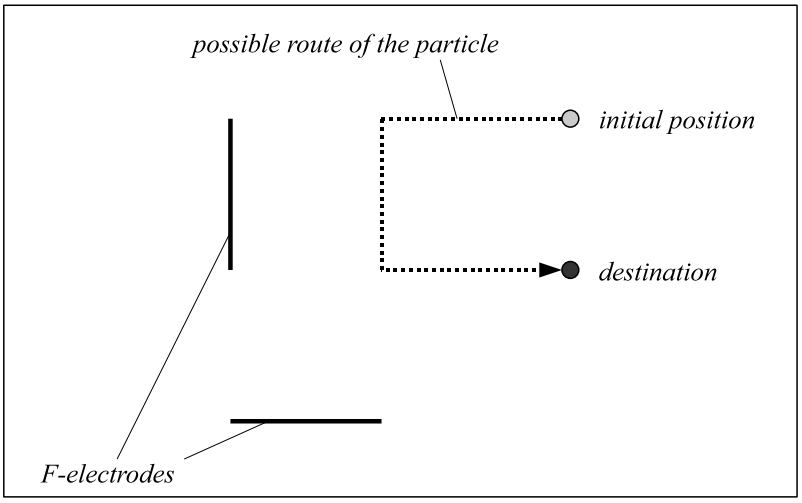
Problem B

Electrophoretic

Input File: electro.in

Scientist Frank, majoring in electrochemistry, has developed line-shaped strange electrodes called *F-electrodes*. During being activated, each F-electrode causes a special potential on and between the two lines touching the F-electrode's endpoints at a right angle. Then electrically-charged particles located inside the potential area get to move in the direction parallel to the potential boundary (i.e. perpendicular to the F-electrode), either toward or against F-electrode. The moving direction can be easily controlled between the two possibles; it is also possible to get particles to pass through F-electrodes. In addition, unlike ordinary electrodes, F-electrodes can affect particles even infinitely far away, as long as those particles are located inside the potential area. On the other hand, two different F-electrodes cannot be activated at a time, since their potentials conflict strongly.

We can move particles on our will by controlling F-electrodes. However, in some cases, we cannot lead them to the desired positions due to the potential areas being limited. To evaluate usefulness of F-electrodes from some aspect, Frank has asked you the following task: to write a program that finds the shortest distances from the particles' initial positions to their destinations with the given sets of F-electrodes.



Input

The input consists of multiple test cases. The first line of each case contains $N(1 \leq N \leq 100)$ which represents the number of F-electrodes. The second line contains four integers x_s, y_s, x_t and y_t, where (x_s, y_s) and (x_t, y_t) indicate the particle's initial position and destination. Then the description of N F-electrodes follow. Each line contains four integers Fx_s, Fy_s, Fx_t and Fy_t, where (Fx_s, Fy_s) and (Fx_t, Fy_t) indicate the two endpoints of an F-electrode. All coordinate values range from 0 to 100 inclusive.

The input is terminated by a case with $N = 0$.

Output

Your program must output the case number followed by the shortest distance between the initial position to the destination. Output "Impossible" (without quotes) as the distance if it is impossible to lead the elementary particle to the destination. Your answers must be printed with five digits after the decimal point. No absolute error in your answers may exceed 10^{-5}.

Sample Input	Output for the Sample Input
2 2 1 2 2 0 0 1 0 0 1 0 2 0	Case 1: 3.00000

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

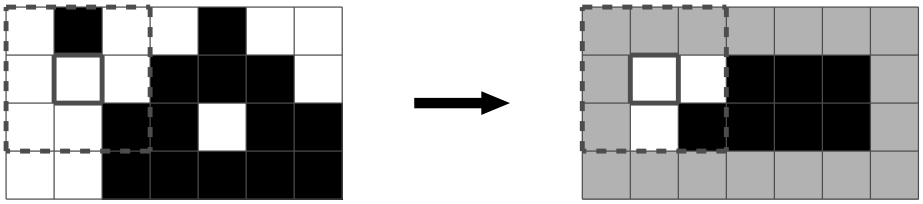
Problem C

Median Filter

Input File: filter.in

The median filter is a nonlinear digital filter used to reduce noise in images, sounds, and other kinds of signals. It examines each sample of the input through a *window* and then emits the *median* of the samples in the window. Roughly speaking, a window is an interval that contains a target sample and its preceding and succeeding samples; the median of a series of values is given by the middle value of the series arranged in ascending (or descending) order.

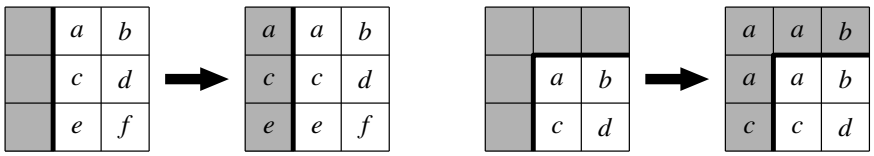
Let us focus on a typical median filter for black-and-white raster images. The typical filter uses a 3×3 window, which contains a target pixel and the eight adjacent pixels. The filter examines each pixel in turn through this 3×3 window, and outputs the median of the nine pixel values, i.e. the fifth lowest (or highest) pixel value, to the corresponding pixel. We should note that the output is just given by the pixel value in majority for black-and-white images, since there are only two possible pixel values (i.e. black and white). The figure below illustrates how the filter works.



before *after*

Note: *The colors of lightly-shaded pixels depend on outside of the region.*

The edges of images need to be specially processed due to lack of the adjacent pixels. In this problem, we extends the original images by repeating pixels on the edges as shown in the figure below. In other words, the lacked pixels take the same values as the nearest available pixels in the original images.



on the left edge *at the left-top corner*

Note: *The letters 'a' through 'f' indicate pixel values.*

You are requested to write a program that reads images to which the filter is applied, then finds the original images containing the greatest and smallest number of black pixels among all possible ones, and reports the difference in the numbers of black pixels.

Input

The input contains a series of test cases.

The first line of each test case contains two integers W and H ($1 \leq W, H \leq 8$), which indicates the width and height of the image respectively. Then H lines follow to describe the filtered image. The i-th line represents the i-th scan line and contains exactly W characters, each of which is either '`#`' (representing black) or '`.`' (representing white).

The input is terminated by a line with two zeros.

Output

For each test case, print a line that contains the case number followed by the difference of black pixels. If there are no original images possible for the given filtered image, print "`Impossible`" instead.

Obey the format as shown in the sample output.

Sample Input	Output for the Sample Input
5 5 ##### ##### ##### ##### ##### 4 4 #### #### #### #### 4 4 #... ....# 4 4 .#.# #.#. .#.# #.#. 0 0	Case 1: 10 Case 2: 6 Case 3: 2 Case 4: Impossible

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem D

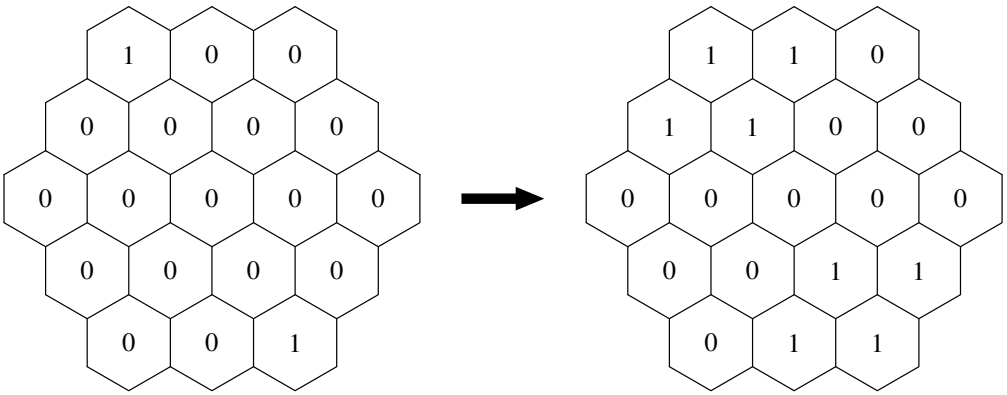
Life Game

Input File: game.in

You are working at a production plant of biological weapons. You are a maintainer of a terrible virus weapon with very high reproductive power. The virus has a tendency to build up regular hexagonal colonies. So as a whole, the virus weapon forms a hexagonal grid, each hexagon being a colony of the virus. The grid itself is in the regular hexagonal form with N colonies on each edge.

The virus self-propagates at a constant speed. Self-propagation is performed simultaneously at all colonies. When it is done, for each colony, the same number of viruses are born at every neighboring colony. Note that, after the self-propagation, if the number of viruses in one colony is more than or equal to the limit density M, then the viruses in the colony start self-attacking, and the number reduces modulo M.

Your task is to calculate the total number of viruses after L periods, given the size N of the hexagonal grid and the initial number of viruses in each of the colonies.



Input

The input consists of multiple test cases.

Each case begins with a line containing three integers N ($1 \leq N \leq 6$), M ($2 \leq M \leq 10^9$), and L ($1 \leq L \leq 10^9$). The following $2N - 1$ lines are the description of the initial state. Each non-negative integer (smaller than M) indicates the initial number of viruses in the colony. The first line contains the number of viruses in the N colonies on the topmost row from left to right, and the second line contains those of $N + 1$ colonies in the next row, and so on.

The end of the input is indicated by a line "0 0 0".

Output

For each test case, output the test case number followed by the total number of viruses in all colonies after L periods.

Sample Input	Output for the Sample Input
3 3 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	Case 1: 8 Case 2: 18

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem E

Subdividing a Land

Input File: land.in

Indigo Real-estate Company is now planning to develop a new housing complex. The entire complex is a square, all of whose edges are equally a meters. The complex contains n subdivided blocks, each of which is a b-meter square. Here both a and b are positive integers.

However the project is facing a big problem. In this country, a percentage limit applies to the subdivision of a land, under the pretext of environmental protection. When developing a complex, the total area of the subdivided blocks must not exceed 50% of the area of the complex; in other words, more than or equal to 50% of the newly developed housing complex must be kept for green space. As a business, a green space exceeding 50% of the total area is a *dead space*. The primary concern of the project is to minimize it.

Of course purchasing and developing a land costs in proportion to its area, so the company also wants to minimize the land area to develop as the secondary concern. You, a member of the project, were assigned this task, but can no longer stand struggling against the problem with your pencil and paper. So you decided to write a program to find the pair of minimum a and b among those which produce the minimum dead space for given n.

Input

The input consists of multiple test cases. Each test case comes in a line, which contains an integer n. You may assume $1 \leq n \leq 10000$.

The end of input is indicated by a line containing a single zero. This line is not a part of the input and should not be processed.

Output

For each test case, output the case number starting from 1 and the pair of minimum a and b as in the sample output.

You may assume both a and b fit into 64-bit signed integers.

Sample Input	Output for the Sample Input
1 2 0	Case 1: 3 2 Case 2: 2 1

Winter Camp for World Finals 2008: Practice Contest I

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

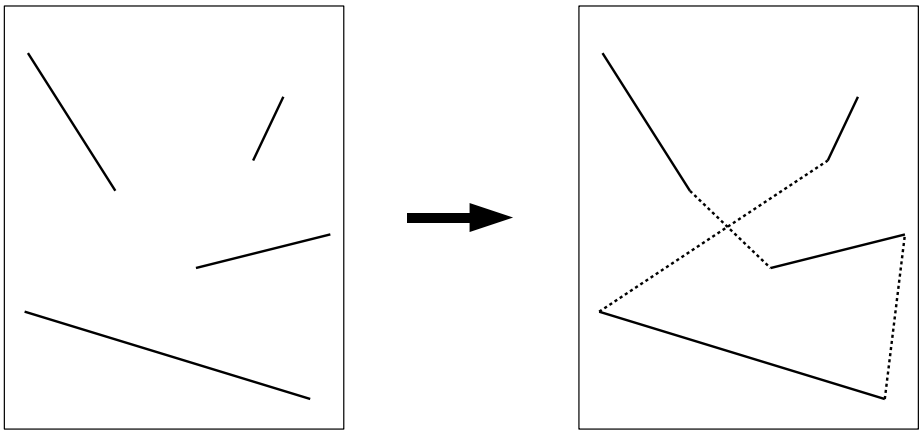
Problem F

Connect Line Segments

Input File: line.in

Your dear son Arnie is addicted to a puzzle named *Connect Line Segments*.

In this puzzle, you are given several line segments placed on a two-dimensional area. You are allowed to add some new line segments each connecting the end points of two existing line segments. The objective is to form a single polyline, by connecting all given line segments, as short as possible. The resulting polyline is allowed to intersect itself.



an initial state

one solution
(*dotted lines added*)

Arnie has solved many instances by his own way, but he is wondering if his solutions are the best one. He knows you are a good programmer, so he asked you to write a computer program with which he can verify his solutions.

Please respond to your dear Arnie's request.

Input

The input consists of multiple test cases.

Each test case begins with a line containing a single integer n ($2 \leq n \leq 14$), which is the number of the initial line segments. The following n lines are the description of the line segments. The i-th line consists of four real numbers: $x_{i,1}$, $y_{i,1}$, $x_{i,2}$, and $y_{i,2}$ ($-100 \leq x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \leq 100$). $(x_{i,1}, y_{i,1})$ and $(x_{i,2}, y_{i,2})$ are the coordinates of the end points of the i-th line segment.

The end of the input is indicated by a line with single "0".

Output

For each test case, output the case number followed by the minimum length in a line.

The output value should be printed with five digits after the decimal point, and should not contain an error greater than 0.00001.

Sample Input	Output for the Sample Input
4 0 1 0 9 10 1 10 9 1 0 9 0 1 10 9 10 2 1.2 3.4 5.6 7.8 5.6 3.4 1.2 7.8 0	Case 1: 36.24264 Case 2: 16.84508

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem G

Oil Company

Input File: oil.in

Irving & Cohen Petroleum Corporation has decided to develop a new oil field in an area. A preliminary survey has been done and they created a detailed grid map of the area which indicates the reserve of oil.

They are now planning to construct mining plants on several grid blocks according this map, but they decided not to place any two plants on adjacent positions to avoid spreading of fire in case of blaze. Two blocks are considered to be adjacent when they have a common edge. You are one of the programmers working for the company and your task is to write a program which calculates the maximum amount of oil they can mine, given the map of the reserve.

Input

The first line of the input specifies N, the number of test cases. Then N test cases follow, each of which looks like the following:

$W\ H$
$r_{1,1}\ r_{2,1}\ \cdots\ r_{W,1}$
$\cdots$
$r_{1,H}\ r_{2,H}\ \cdots\ r_{W,H}$

The first line of a test case contains two integers W and H ($1 \leq W, H \leq 20$). They specifies the dimension of the area. The next H lines, each of which contains W integers, represent the map of the area. Each integer $r_{x,y}$ ($0 \leq r_{x,y} < 10000$) indicates the oil reserve at the grid block (x, y).

Output

For each test case, output the case number (starting from 1) and the maximum possible amount of mining in a line. Refer to the sample output section about the format.

Sample Input	Output for the Sample Input
2 2 2 2 3 3 5 3 2 4 1 1 2 1 4	Case 1: 7 Case 2: 8

Winter Camp for World Finals 2008: Practice Contest I

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem H

Finding the Top RPS Player

Input File: rps.in

A company "ACM Foods" is preparing for opening its chain shop in a certain area, but another company "ICPC Pizza" is also planning to set up its branch shop in the same area. In general, two competitive shops gain less incomes if they are located so close to each other. Thus, if both "ACM Foods" and "ICPC Pizza" went on opening, they would be damaged financially. So, they had a discussion on this matter and made the following agreement: only one of them can branch its shop in the area. It is determined by Rock-Paper-Scissors (RPS) which to branch the shop.

ACM Foods is facing financial difficulties and strongly desires to open their new shop in that area. The executives have decided to make every effort for finding out a very strong RPS player. They believes that players who win consecutive victories must be strong players. In order to find such a player for sure, they have decided their simple strategy.

In this strategy, many players play games of RPS repeatedly, but the games are only played between players with the same number of consecutive wins. At the beginning, all the players have no wins, so any pair of players can play a game. The games can be played by an arbitrary number of pairs simultaneously. Let us call a set of simultaneous games as a *turn*. After the first turn, some players will have one win, and the other players will remain with no wins. In the second turn, some games will be played among players with one win, and some other games among players with no wins. For the former games, the winners will have two consecutive wins, and the losers will lose their first wins and have no *consecutive* wins. For the latter games, the winners will have one win, and the losers will remain with no wins. Therefore, after the second turn, the players will be divided into three groups: players with two consecutive wins, players with one win, and players with no wins. Again, in the third turn, games will be played among players with two wins, among with one win, and among with no wins. The following turns will be conducted so forth. After a sufficient number of turns, there should be a player with the desired number of consecutive wins.

The strategy looks crazy? Oh well, maybe they are confused because of their financial difficulties.

Of course, this strategy requires an enormous amount of plays. The executives asked you, as an employee of ACM Foods, to estimate how long the strategy takes. Your task is to write a program to count the minimum number of turns required to find a player with M consecutive wins among N players.

Input

The input consists of multiple test cases. Each test case consists of two integers N ($2 \leq N \leq 20$) and M ($1 \leq M < N$) in one line.

The input is terminated by the line containing two zeroes.

Output

For each test case, your program must output the case number followed by one integer which indicates the minimum number of turns required to find a person with M consecutive wins.

Sample Input	Output for the Sample Input
2 1 10 5 15 10 0 0	Case 1: 1 Case 2: 11 Case 3: 210

Winter Camp for World Finals 2008: Practice Contest I

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem I

Revenge of Voronoi

Input File: voronoi.in

A discrete Voronoi diagram is a derivation of a Voronoi diagram. It is represented as a set of pixels. Each of the generatrices lies on the center of some pixel. Each pixel belongs to the generatrix nearest from the center of the pixel in the sense of Manhattan distance. The Manhattan distance d between two points (x_1, y_1) and (x_2, y_2) is given by the following formula:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Your task is to find a set of generatrices which generates a given discrete Voronoi diagram. In the given diagram, each generatrix is given a unique lowercase letter as its identifier, and each pixel is represented by the identifier of the generatrix the pixel belongs to. If a pixel has multiple generatrices at the same distance from its center, it belongs to the generatrix with the most preceding identifier among them (i.e. the smallest character code).

Input

The input consists of multiple test cases.

Each test case begins with a line containing two integers W ($1 \leq W \leq 32$) and H ($1 \leq H \leq 32$), which denote the width and height of the discrete Voronoi diagram.

The following H lines, each of which consists of W letters, give one discrete Voronoi diagram. Each letter represents one pixel.

The end of input is indicated by a line with two zeros. This is not a part of any test cases.

Output

For each test case, print the case number and the coordinates of generatrices as shown in the sample output. Each generatrix line should consist of its identifier, x-coordinate, and y-coordinate. Generatrices should be printed in alphabetical order of the identifiers. Each coordinate is zero-based where $(0, 0)$ indicates the center of the top-left corner pixel of the diagram.

You may assume that every test case has at least one solution. If there are multiple solutions, any one is acceptable.

Print a blank line after every test case including the last one.

Sample Input

```
4 3
ooxx
ooxx
ooxx
4 1
null
4 4
aabb
aabb
ccdd
ccdd
0 0
```

Output for the Sample Input

```
Case 1:
o 0 0
x 2 0

Case 2:
l 2 0
n 0 0
u 1 0

Case 3:
a 0 0
b 2 0
c 0 2
d 2 2
```

Winter Camp for World Finals 2008: Practice Contest I

Winter Camp for ACM-ICPC World Finals 2008

presented by JAG

Practice Contest I

Problem J

Castle Wall

Input File: wall.in

A new lord assumed the position by the death of the previous lord in a Far Eastern province.

The new greedy lord hates concave polygons, because he believes they need much wasted area to be drawn on paper. He always wants to modify them to convex ones.

His castle is currently surrounded by a wall forming a concave polygon, when seen from the above. Of course he hates it. He believes more area could be obtained with a wall of a convex polygon. Thus he has ordered his vassals to have new walls built so they form a convex polygon.

Unfortunately, there is a limit in the budget. So it might be infeasible to have the new walls built completely. The vassals has found out that only up to r meters of walls in total can be built within the budget. In addition, the new walls must be built in such a way they connect the polygonal vertices of the present castle wall. It is impossible to build both of intersecting walls.

After long persuasion of the vassals, the new lord has reluctantly accepted that the new walls might not be built completely. However, the vassals still want to maximize the area enclosed with the present and new castle walls, so they can satisfy the lord as much as possible.

Your job is to write a program to calculate, for a given integer r, the maximum possible area of the castle with the new walls.

Input

The input file contains several test cases.

Each case begins with a line containing two positive integers n and r. n is the number of vertices of the concave polygon that describes the present castle wall, satisfying $5 \leq n \leq 64$. r is the maximum total length of new castle walls feasible within the budget, satisfying $0 \leq r \leq 400$.

The subsequent n lines are the x- and y-coordinates of the n vertices. The line segments (x_i, y_i)–(x_{i+1}, y_{i+1}) $(1 \leq i \leq n-1)$ and (x_n, y_n)–(x_1, y_1) form the present castle wall of the concave polygon. Those coordinates are given in meters and in the counterclockwise order of the vertices.

All coordinate values are integers between 0 and 100, inclusive. You can assume that the concave polygon is simple, that is, the present castle wall never crosses or touches itself.

The last test case is followed by a line containing two zeros.

Output

For each test case in the input, print the case number (beginning with 1) and the maximum possible area enclosed with the present and new castle walls. The area should be printed with exactly one fractional digit.

Sample Input	Output for the Sample Input
5 4 0 0 4 0 4 4 2 2 0 4 8 80 45 41 70 31 86 61 72 64 80 79 40 80 8 94 28 22 0 0	case 1: 16.0 case 2: 3375.0