

Problem A

2025

Time Limit: 2 seconds

In Japan, the multiplication table below is learned in elementary schools.

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

Such a table with 9×9 items, called *kuku* in Japanese, is not necessarily a world standard. In some cultures, 12×12 or 20×20 tables are learned.

Interestingly, the total of all the 9×9 items in the *kuku* table is 2025, which happens to be this year. Are there any other interesting cases like this?

Your task is to compute, for a given integer n , the sum of all the items in an $n \times n$ multiplication table. More specifically, find

$$\sum_{a=1}^n \sum_{b=1}^n ab.$$

Input

The input consists of at most 100 test cases. Each test case is a line containing one integer n ($1 \leq n \leq 100$).

The end of the input is indicated by a line consisting only of a zero.

Output

For each test case, output the value of

$$\sum_{a=1}^n \sum_{b=1}^n ab$$

in a line.

Sample inputs and outputs are available on the [Problemset page in DOMjudge](#).



Sample Input 1

Sample Output 1

9	2025
1	1
2	9
100	25502500
0	

Problem B

Prefix and Suffix Can Be the Same

Time Limit: 2 seconds

In this problem, a prefix of a string means a consecutive substring starting from its first character. For example, the prefixes of the string `icpc` are `i`, `ic`, `icp`, and `icpc`. Similarly, a suffix of a string means a consecutive substring ending at its last character. For example, the suffixes of the string `icpc` are `icpc`, `cpc`, `pc`, and `c`.

Given a string s , find the shortest string other than s itself having s as both its prefix and its suffix. It can be proven that such a string is unique.

Input

The input consists of at most 50 test cases, each in the following format.

```
 $n$   
 $s$ 
```

The first line contains an integer n between 1 and 50, inclusive. The second line contains a string s of length n , consisting of English lowercase letters.

The end of the input is indicated by a line consisting only of a zero.

Output

For each test case, output in a line the shortest string other than s having s as both its prefix and its suffix.

Sample inputs and outputs are available on the [Problemset page in DOMjudge](#).

Sample Input 1

```
4  
test  
4  
icpc  
8  
cccpppcc  
1  
i  
9  
strongest  
3  
ttt  
0
```

Sample Output 1

```
testest  
icpcicpc  
cccpppccpppcc  
ii  
strongstrongest  
tttt
```

Problem C

Calendar of an Enthusiastic Worker

Time Limit: 2 seconds

You are an enthusiastic worker. You want to know how many days you can work during the next specified number of days.

Saturdays and Sundays are non-working days. In addition, a number of national holidays and some company-specific days are non-working days. You cannot work on non-working days. Such non-working days may overlap: The founding anniversary of the company may coincide with a national holiday. Even when such non-working days coincide, there is no system to grant additional non-working days.

Today is Monday. Your task is to count the number of days you can work within a specified period starting from today.

Input

The input consists of at most 100 test cases. Each test case is given in the following format.

```
n m
a1 ⋯ an
```

The first line contains the number of additional non-working days n , and the number of days of the specified period m ($1 \leq n \leq 300$, $1 \leq m \leq 10^{18}$). The second line contains n integers, a_1, \dots, a_n ($1 \leq a_i \leq 10^{18}$). With today being the first day, the a_i -th day for $i = 1, \dots, n$ is a non-working day. The list a_1, \dots, a_n may contain duplicates.

The end of the input is indicated by a line consisting of two zeros.

Output

For each test case, output in a line the number of days you can work during m days starting from today.

Sample inputs and outputs are available on the [Problemset page in DOMjudge](#).

Sample Input 1	Sample Output 1
4 8	2
1 2 3 4	7
2 10	10
15 3	714285714285714282
3 14	2
6 7 13	
5 10000000000000000000	
8 15 1 8 999999999999999994	
2 3	
1 4	
0 0	

Problem D

Ancient Game Board

Time Limit: 2 seconds

In ancient *Icpca* civilization, games were played on a board covered with square tiles of the same size. It is certain that the tiles had either black or white surfaces, but their arrangement is not yet known.

One of the most influential theories says that the board was divided into square regions of the same size, each filled with the tiles of a single color, either black or white. It says that the regions had alternating colors, in both front-back and left-right directions, like a chessboard.

Today, a rectangular part of the board used in the game was discovered. Your task is to determine whether this artifact is consistent with the theory described above. If it is consistent, also determine whether the number of tiles as the side length of the square regions can be uniquely identified.

The artifact of the first test case of Sample Input 1 is shown on the left of Figure D.1. This artifact can be considered as part of a board where the square regions have sides of 2 tiles, as shown in the red frame on the right, and thus the artifact is consistent with the theory. Furthermore, since it cannot be a part of a board with square regions of any other size, the side length is uniquely identified to be 2 tiles.

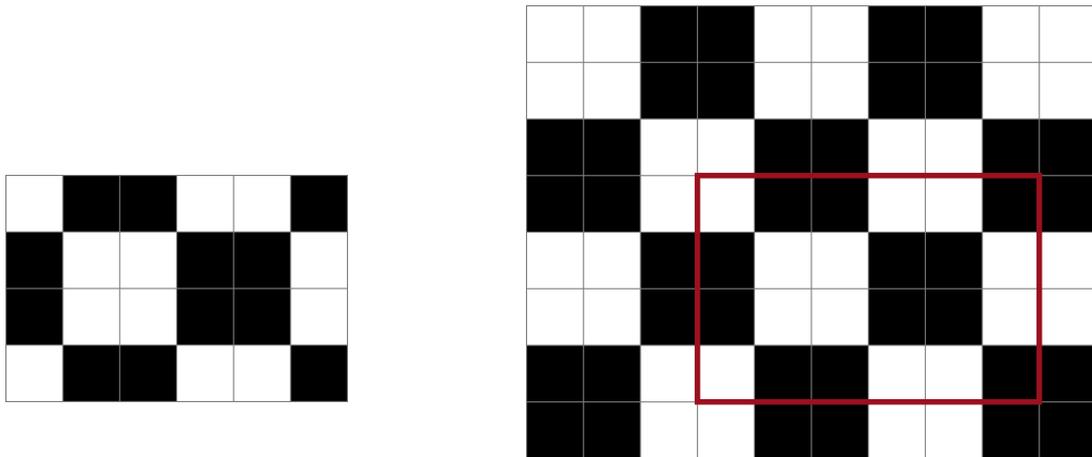


Figure D.1. The first test case of Sample Input 1

Input

The input consists of at most 30 test cases, each in the following format.

```

n m
c1,1 ··· c1,m
⋮
cn,1 ··· cn,m

```

The first line of the test case contains two integers, n and m ($1 \leq n \leq 100, 1 \leq m \leq 100$), representing the numbers of tile rows and columns, respectively, of the artifact (i.e., the remaining part of a board). Each of the next n lines contains a string of m characters, each being ‘.’ or ‘#’, representing the colors of the tiles in the artifact. The j -th character in the i -th line of the n lines $c_{i,j}$ being ‘.’ means that the j -th tile of the i -th row of the artifact is white, and being ‘#’ means it is black ($1 \leq i \leq n, 1 \leq j \leq m$).

The end of the input is indicated by a line consisting of two zeros.

Output

For each test case, output one of the following integers:

- An integer -1 , if the artifact is not consistent with the theory.
- The number of tiles as the side length of the square regions, if the artifact is consistent with the theory and the side length can be uniquely identified.
- An integer 0 , if the artifact is consistent with the theory but the number of tiles as the side length of the square regions cannot be uniquely identified.

Sample inputs and outputs are available on the [Problemset](#) page in DOMjudge.

Sample Input 1	Sample Output 1
4 6 .##..# #..##. #..##. .##..#	2 -1 0 -1 4
3 3 ### #.# ###	1
2 3 #.. .##	
4 3 #.# #.# #.# #.#	
4 6 .####. #....# #....# #....#	
3 3 #.# .#. #.#	
0 0	

Problem E

To Be Discontinued

Time Limit: 2 seconds

The year is AD 3000. Humanity enjoys traveling among the stars via the network of interstellar spaceship routes. Due to recent advances in warp drive technology, however, the traditional spaceship routes are being gradually discontinued. You, a nostalgic voyager, have decided to take all the remaining routes before the planned discontinuation.

Currently, the interstellar routes form a tree-structured network among n planets. That is, there are $n - 1$ bidirectional routes each connecting two planets, and by taking such routes several times, you can travel between any pair of planets. However, each route has now been assigned its final boarding time. If you are on a planet at time t , and t does not exceed the final boarding time e of a certain route starting there (i.e., $t \leq e$), you can reach the other end of the route at time $t + 1$. You cannot use the route after the final boarding time.

In addition to the traditional spaceship routes, you can also use the new warp drive technology. Using the warp drive, you can instantly travel from any planet to a destination planet without spending any time, provided that you have already visited that destination at least once! You are currently on one of the n planets. Because you have not visited any other planets prior to this journey, you must first travel to a planet via the spaceship routes before you can warp to it.

Starting your journey at time 0, can you traverse all the spaceship routes before their respective final boarding times? Since the network forms a tree, the question is equivalent to asking whether you can visit all the planets. If such a journey is possible, find the order in which you should visit the planets.

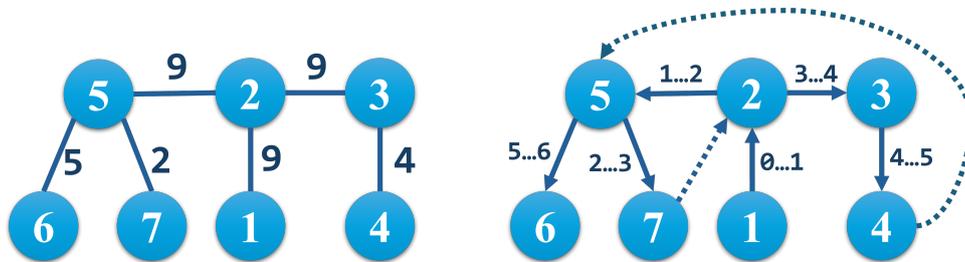


Figure E.1. The first test case of Sample Input 1

Figure E.1 illustrates the first test case of Sample Input 1. The left part shows the network of interstellar spaceship routes. Each line segment is labeled with an integer representing the final boarding time of the route. The right part shows one way to traverse all the routes. Dotted lines in the right part represent warps, and the notation “ $t \dots t + 1$ ” represents taking the route that departs at time t and arrives at time $t + 1$.

Input

The input consists of at most 50 test cases, each in the following format.

```

n
p2 e2
⋮
pn en

```

The integer n ($2 \leq n \leq 1000$) on the first line is the number of planets connected by the network of interstellar spaceship routes. The planets are numbered 1 through n and you are initially on planet 1. The 2-nd to the n -th lines describe the interstellar spaceship routes, where integers p_i ($1 \leq p_i < i$) and e_i ($0 \leq e_i \leq 1000$) mean that there exists a route between planets i and p_i with the final boarding time e_i , for $i = 2, \dots, n$.

The end of the input is indicated by a line consisting only of a zero.

Output

For each test case, if it is possible to traverse all the routes meeting their respective final boarding times, output `yes` in the first line; otherwise, output `no`.

If that is possible, output in the next line a list of the planet numbers from 2 to n in the order of your first visits, separated by a space. If there are multiple ways, output any one of them.

Sample inputs and outputs are available on the [Problemset page in DOMjudge](#).

Sample Input 1

```
7
1 9
2 9
3 4
2 9
5 5
5 2
4
1 1
1 1
1 1
0
```

Sample Output 1

```
yes
2 5 7 3 4 6
no
```

Problem F

Dog Tricks

Time Limit: 2 seconds

The videos showcasing a wide variety of tricks performed by your two pet dogs, Amy and Bessie, have become popular content in the Icpca Kingdom. To explore the potential for further videos, you taught the dogs a new trick.

The new trick is performed using plates aligned from left to right. Each plate holds either an apple or a banana.

When you command Amy, she holds a banana in her mouth and looks at each plate in turn from left to right. She skips any plate with a banana and, upon reaching the first plate with an apple, places the banana she was carrying on that plate, picks up the apple, and continues to the right. Then she skips any plate with an apple and, upon reaching the first plate with a banana, places the apple she was carrying on that plate, picks up the banana, and comes back to you, successfully completing the trick. If the fruits on the plates are not arranged in a way that allows this, the trick fails.

When you command Bessie, she holds an apple in her mouth and looks at each plate in turn from left to right. She skips any plate with an apple and, upon reaching the first plate with a banana, places the apple she was carrying on that plate, picks up the banana, and continues to the right. Then she skips any plate with a banana and, upon reaching the first plate with an apple, places the banana she was carrying on that plate, picks up the apple, and comes back to you, successfully completing the trick. If the fruits on the plates are not arranged in a way that allows this, the trick fails.

You cannot command both dogs simultaneously, nor can you command one dog while the other is performing a trick.

In the next video project, you want to rearrange the fruits on the plates into the target state by repeatedly having the dogs perform successful tricks. You are given the initial and target states of the fruits on the plates. Starting from the initial state, can you rearrange the fruits on the plates into the target state by issuing no more than a specified number of commands to Amy and Bessie in an appropriate order? If possible, find a sequence of such commands.

Input

The input consists of at most 100 test cases, each in the following format.

n
 s
 t

Each test case consists of three lines. The first line contains an integer n between 1 and 100, inclusive, representing the number of plates. The second line contains a string s representing the initial state. If the i -th character of s is `a`, an apple is on the i -th plate from the left in the initial state; if it is `b`, a banana is on it. The third line contains a string t representing the target state. If the i -th character of t is `a`, an apple should be on the i -th plate from the left in the target state; if it is `b`, a banana should be on it. Both s and t consist only of `a` and `b`, and they both have length n . Strings s and t are not identical.

The end of the input is indicated by a line consisting only of a zero.

Output

For each test case, output `yes` in a line if there exists a sequence of 10 000 or less commands to the dogs that rearranges the initial state to the target state; otherwise, output `no` in a line. In addition, if such a sequence exists, output a string representing the command sequence in the next line. The string should consist of the characters `A` and `B`. Its i -th character being `A` means the i -th command is to Amy, while its being `B` means the command is to Bessie. If there are two or more such sequences, any of them is accepted.

Sample inputs and outputs are available on the [Problemset page in DOMjudge](#).

Sample Input 1	Sample Output 1
2	no
aa	yes
bb	BA
3	yes
bba	AB
bab	
4	
aaba	
abaa	
0	

In the second test case of Sample Input 1, starting from the initial state `bba`, you first command Bessie. She replaces the leftmost banana with an apple, resulting in `aba`, and then she replaces the rightmost apple with a banana, resulting in `abb`. Next, you command Amy. She replaces the leftmost apple with a banana, resulting in `bbb`, and then she replaces the next banana with an apple, resulting in `bab`. This matches the target state.

Problem G

Number of Faces

Time Limit: 2 seconds

Two planes, H_1 and H_2 , are in a three-dimensional Euclidean space with axes, x , y , and z , where H_1 is defined by $z = 1$ and H_2 by $z = 2$.

You are given n real numbers, d_1, \dots, d_n , and m real numbers, d'_1, \dots, d'_m . These real numbers are positive and strictly less than 180. Consider drawing the following convex polygons on the planes H_1 and H_2 .

- On H_1 , you draw an n -sided polygon. The interior angles at its vertices are d_1, \dots, d_n degrees in counterclockwise order as viewed from the origin.
- Similarly, on H_2 , you draw an m -sided polygon. The interior angles at its vertices are d'_1, \dots, d'_m degrees in counterclockwise order as viewed from the origin.

Here, only the interior angles of the polygons are specified; the lengths of their edges and the positions of their vertices are not.

Once the positions of the two polygons are fixed, the convex polyhedron whose vertex set is these $n + m$ vertices is uniquely determined. Write a program that enumerates all the possible numbers of faces that such a convex polyhedron can have.

Here, all the dihedral angles (angles between two adjacent faces) of a convex polyhedron must be strictly less than 180 degrees.

In the first test case of Sample Input 1, quadrilaterals whose interior angles are all 90 degrees are drawn on H_1 and H_2 . For example, a rectangular cuboid can be constructed as in Figure G.1 (a), which has six faces. By rotating one of the quadrilaterals as shown in Figure G.1 (b), a convex polyhedron with ten faces can be constructed. The possible numbers of faces are six and ten.

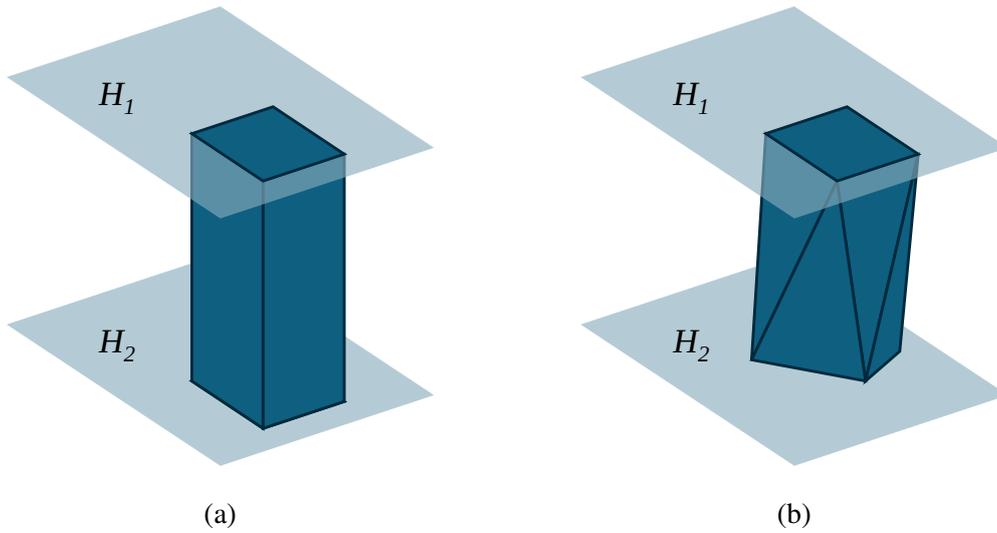


Figure G.1. The first test case of Sample Input 1

Input

The input consists of at most 50 test cases, each in the following format.

n
 d_1
 \vdots
 d_n
 m
 d'_1
 \vdots
 d'_m

The integer n represents the number of vertices of the polygon drawn on H_1 ($3 \leq n \leq 50$). The real numbers, d_1, \dots, d_n , represent the interior angles. They are at least 10^{-9} and strictly less than 180, and are given with exactly nine digits after the decimal point. They satisfy $d_1 + \dots + d_n = (n - 2) \times 180$.

Similarly, the integer m represents the number of vertices of the polygon drawn on H_2 ($3 \leq m \leq 50$). The real numbers, d'_1, \dots, d'_m , represent the interior angles. They are at least 10^{-9} and strictly less than 180, and are given with exactly nine digits after the decimal point. They satisfy $d'_1 + \dots + d'_m = (m - 2) \times 180$.

The end of the input is indicated by a line consisting only of a zero.

Output

For each test case, output in a line all possible numbers of faces that the convex polyhedron can have, in ascending order, separated by a space.

Sample inputs and outputs are available on the Problemset page in DOMjudge.



Sample Input 1

Sample Output 1

```
4
90.000000000
90.000000000
90.000000000
90.000000000
4
90.000000000
90.000000000
90.000000000
90.000000000
3
33.333333333
66.666666666
80.000000001
3
80.000000001
66.666666666
33.333333333
3
59.165980540
68.504848124
52.329171336
5
87.702342452
144.626828884
97.879972796
169.296126888
40.494728980
0
```

```
6 10
6 7 8
8 9 10
```

Problem H

Parentheses

Time Limit: 2 seconds

A number of stamps are lined up from left to right in front of you, each engraved with an open parenthesis, ‘(’, or a close parenthesis, ‘)’. First, you choose one of them and press it onto a piece of paper. The stamp is then returned to its original position. Next, you choose the stamp immediately to the left or right of the one you just returned, and press it directly next to the previous print. You repeat this operation of selecting either the left or right stamp and pressing it. During this process, you may select the stamp at the same position any number of times, but you may not select it consecutively.

A line consisting only of open and close parentheses will be printed on the paper. The printed line is valid if, counting characters in the line one by one from its start, the number of open parentheses is always greater than or equal to that of the close parentheses, and the two counts are equal at the end. Find the number of pairs of first and last stamps with which you can print a valid line by appropriately selecting stamps in the process.

Input

The input consists of at most 10^5 test cases, each in the following format.

n
 s

A test case consists of two lines. The first line contains an integer n representing the number of stamps ($2 \leq n \leq 2 \times 10^5$). The second line contains a string s of n characters, consisting of ‘(’ or ‘)’. Its i -th character describes the character of the i -th stamp from the left.

The end of the input is indicated by a line containing a zero. The sum of n over all the test cases does not exceed 2×10^5 .

Output

For each test case, output in a line the number of pairs that satisfy the condition.

Sample inputs and outputs are available on the [Problemset](#) page in DOMjudge.

Sample Input 1
Sample Output 1

5	3
))) ((9
10	4 6
) ()) (()) (
20	
(() ((())))) () () (
0	

In the first test case of Sample Input 1, suppose that the stamps are numbered from left to right, with the leftmost stamp numbered 1. The pairs of first and last stamps that satisfy the condition are $(4, 1)$, $(4, 3)$, and $(5, 2)$. For example, for the pair $(4, 1)$, selecting stamps in the order 4, 5, 4, 3, 2, 1 yields the printed line “((()))”.

Problem I

Preparing the Lunch

Time Limit: 2 seconds

Students participating in the ICPC are sitting evenly spaced around a round table for lunch. Each student has one catered lunch box. There are various types of lunch boxes, and not all students necessarily have the same type. The number of the students is even. Thus, for each student, there is exactly one student on the directly opposite seat of the table. The number of lunch boxes of each type is even.

As the lunch box on the seat directly opposite is easily visible, a student may feel envious if it is not of the same type as that student's own. Therefore, you want each pair of students sitting opposite each other to have the same type of lunch boxes. As the number of lunch boxes of each type is even, such an arrangement is always possible. To rearrange the lunch boxes, you repeat the following operation any number of times: select any pair of adjacent students and have them exchange their lunch boxes. Find the minimum number of such operations required to achieve the goal.

Input

The input consists of at most 5×10^4 test cases, each in the following format.

```

n
a1 ··· a2n

```

Each test case consists of two lines. The first line contains an integer n ($2 \leq n \leq 2 \times 10^5$), meaning that there are $2n$ students at the lunch table. The second line contains $2n$ integers a_1, \dots, a_{2n} . The i -th integer a_i denotes the type of lunch box initially held by the i -th student ($i = 1, \dots, 2n$), where the students are numbered clockwise around the table from a specific position. Each integer a_i is between 1 and n , inclusive. For any integer j , the number of indices i such that $a_i = j$ is guaranteed to be even.

The end of the input is indicated by a line consisting only of a zero. The sum of n over all the test cases does not exceed 2×10^5 .

Output

For each test case, output in a line the minimum number of operations required to achieve the goal.

Sample inputs and outputs are available on the [Problemset page in DOMjudge](#).

Sample Input 1	Sample Output 1
<pre>4 4 2 3 4 1 2 1 3 4 1 2 1 3 2 2 2 3 3 3 3 3 3 3 3 0</pre>	<pre>2 3 0</pre>

For the first test case, swapping the 1 at the seventh position with the 3 in the next position results in 4 2 3 4 1 2 3 1, and then swapping the 1 at the eighth position with the 4 at the first position results in



1 2 3 4 1 2 3 4. After these two swaps, all the pairs of lunch boxes at the direct opposite seats are of the same type. Since the goal cannot be achieved with a single swap, the answer is 2.