KTH training 2025-11-13

A. Year 2025

2 s., 1024 MB

	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

This is an example of a multiplication table of 9×9 items.

Interestingly, the sum of all the 9×9 items 2025, which happens to be this year. You wonder, 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, compute

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

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.

input

```
9
1
2
100
0
```

output

```
2025
1
9
25502500
```

B. Prefix and Suffix Can Be the Same

2 seconds, 1024 megabytes

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.

input

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

output

```
testest
icpcicpc
cccpppcccpppcc
ii
strongestrongest
tttt
```

C. Empty Triangle

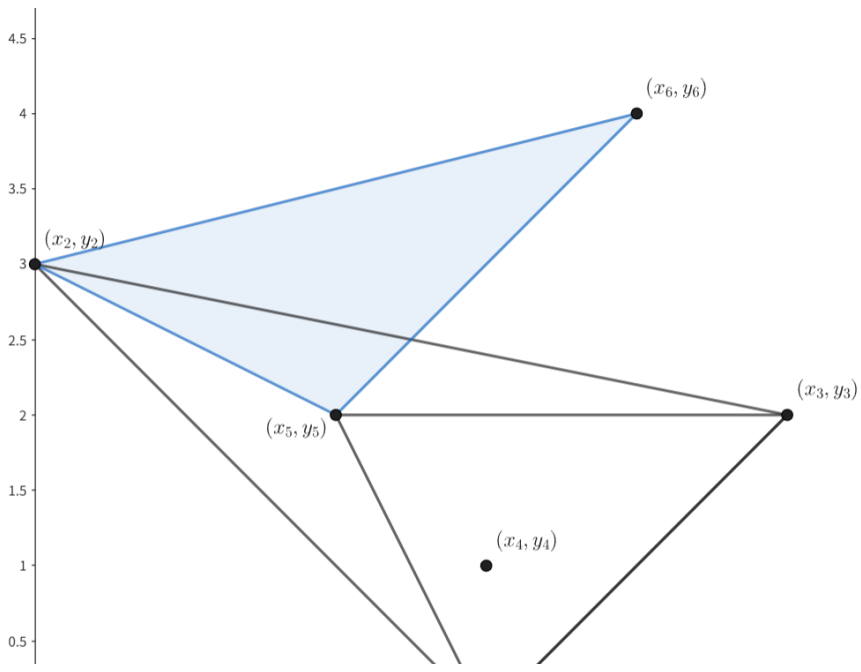
2 seconds, 512 megabytes

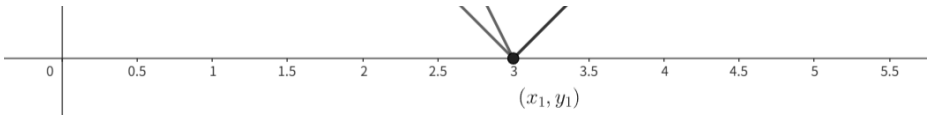
This is an interactive problem.

The pink soldiers hid from you n $(3 \leq n \leq 1500)$ **fixed** points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, **whose coordinates are not given to you**. Also, it is known that no two points have the same coordinates, and no three points are collinear.

You can ask the Frontman about three **distinct** indices i, j, k. Then, he will draw a triangle with points (x_i, y_i), (x_j, y_j), (x_k, y_k), and respond with the following:

- If at least one of the hidden points lies inside the triangle, then the Frontman gives you the index of one such point. Do note that if there are multiple such points, **the Frontman can arbitrarily select one of them**.
- Otherwise, the Frontman responds with 0.





Your objective in this problem is to find a triangle not containing any other hidden point, such as the blue one in the diagram.

Using at most $\mathbf{75}$ queries, you must find any triangle formed by three of the points, **not containing** any other hidden point inside.

Do note that the Frontman may be **adaptive** while choosing the point to give you. In other words, the choice of the point can be determined by various factors including but not limited to the orientation of points and the previous queries. However, note that **the sequence of points will never be altered**.

Hacks are disabled for this problem. Your solution will be judged on exactly $\mathbf{35}$ input files, including the example input.

Interaction

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 20$). The description of the test cases follows.

The first line of each test case contains one positive integer n — the number of points ($3 \le n \le 1500$).

Afterwards, you can output the following on a separate line to ask a query:

- `"? i j k"` ($1 \le i, j, k \le n, i \ne j, i \ne k, j \ne k$): Ask about the triangle formed by (x_i, y_i), (x_j, y_j), (x_k, y_k).

Then, the interactor responds with one integer on a new line:

- If your query is invalid or the number of queries has exceeded $\mathbf{75}$, then the interactor responds with -1;
- If there exists an index p ($1 \le p \le n$) such that the point (x_p, y_p) is inside the triangle, then **any such index** p is given;
- Otherwise, the interactor responds with 0.

If you want to print an answer, you can output the following on a separate line:

- `"! i j k"` ($1 \le i, j, k \le n, i \ne j, i \ne k, j \ne k$): The triangle formed by (x_i, y_i), (x_j, y_j), (x_k, y_k) contains no other hidden point.

Then, the following interaction happens:

- If the answer is invalid or the triangle contains another hidden point, the interactor responds with -1;
- If the answer is correct and there are more test cases to solve, you are given the value of n for the next test case;
- Otherwise, it means that all test cases are solved correctly, and your program may terminate gracefully.

Do note that printing the answer does **not** count towards the number of queries made.

Do note that the interactor may be **adaptive** while choosing the point to give you. In other words, the choice of the point can be determined by various factors including but not limited to the orientation of points and the previous queries. However, note that **the sequence of points will never be altered**.

After printing each query do not forget to output the end of line and flush* the output. Otherwise, you will get `Idleness limit exceeded` verdict.

If, at any interaction step, you read -1 instead of valid data, your solution must exit immediately. This means that your solution will receive `Wrong answer` because of an invalid query or any other mistake. Failing to exit can result in an arbitrary verdict because your solution will continue to read from a closed stream.

Hacks

Hacks are disabled for this problem. Your solution will be judged on exactly $\mathbf{35}$ input files, including the example input.

*To flush, use:

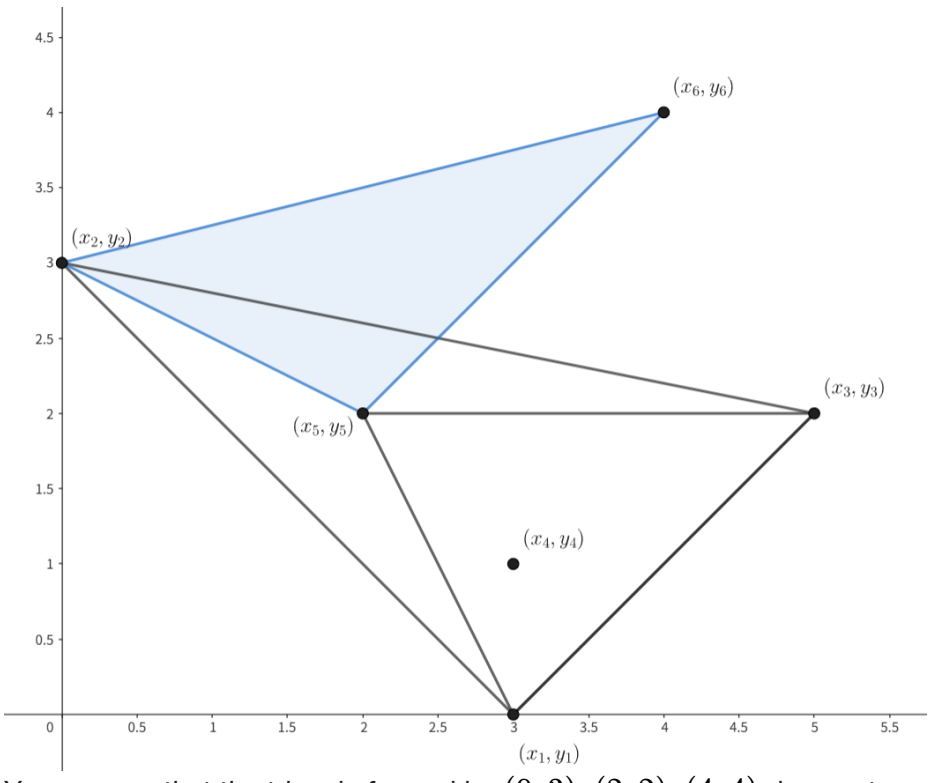
- `fflush(stdout)` or `cout.flush()` in C++;
- `sys.stdout.flush()` in Python;
- see the documentation for other languages.

input
2 6 5 4 0 3

output
  ? 1 2 3 ? 1 5 3 ? 2 5 6 ! 2 5 6 ! 1 2 3

In the first test case, the points are $(3, 0), (0, 3), (5, 2), (3, 1), (2, 2), (4, 4)$.

The triangles corresponding to the three queries are as follows.



You can see that the triangle formed by $(0, 3)$, $(2, 2)$, $(4, 4)$ does not contain any other hidden point inside. Therefore, it is a valid answer.

Do note that the interaction example only shows a valid interaction, and **not necessarily the actual response**. For example, it is possible that the Frontman responds with 4 when you query `"? 1 2 3"`. However, as the Frontman does not alter the sequence of points, he never responds with 6 for the same query.

In the second test case, there are only 3 points. Therefore, we know that the unique triangle formed by the points does not contain any other hidden point inside.

D. Parentheses

2 seconds, 1024 megabytes

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.

input
5)))((10)())(())(20 ((()(((((((()()()(0

output
3 9 46

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 "`( ( ( ) ) )`".

E. Taxi Cab Scheme

1 second, 256 megabytes

Running a taxi station is not all that simple. Apart from the obvious demand for a centralised coordination of the cabs in order to pick up the customers calling to get a cab as soon as possible, there is also a need to schedule all the taxi rides which have been booked in advance. Given a list of all booked taxi rides for the next day, you want to minimise the number of cabs needed to carry out all of the rides.

For the sake of simplicity, we model a city as a rectangular grid. An address in the city is denoted by two integers: the street and avenue number. The time needed to get from the address a, b to c, d by taxi is $|a - c| + |b - d|$ minutes. A cab may carry out a booked ride if it is its first ride of the day, or if it can get to the source address of the new ride from its latest, at least one minute before the new ride's scheduled departure. Note that some rides may end after midnight.

Input

On the first line of the input is a single positive integer N, telling the number of test scenarios to follow. Each scenario begins with a line containing an integer M, $0 < M < 500$, being the number of booked taxi rides. The following M lines contain the rides. Each ride is described by a departure time on the format `hh:mm` (ranging from `00:00` to `23:59`), two integers a b that are the coordinates of the source address and two integers c d that are the coordinates of the destination address. All coordinates are at least 0 and strictly smaller than 200. The booked rides in each scenario are sorted in order of increasing departure time.

Output

For each scenario, output one line containing the minimum number of cabs required to carry out all the booked taxi rides.

input
2 2 08:00 10 11 9 16 08:07 9 16 10 11 2 08:00 10 11 9 16 08:06 9 16 10 11

output
1 2

F. Word Encoding

1 second, 256 megabytes

In any language, certain combinations of letters do not appear (or at least appear so seldom that they can be considered non-existent). For instance, there are no English words containing the three letter combination `buv` as a substring. Given a list of letter combinations that do not exist, the number of possible "words" in a language can be reduced a lot (a "word" here means any combination of letters that doesn't contain any of the given letter combinations as a substring). If we order all such words by increasing length, ordering words of the same length alphabetically, we can enumerate them starting from 1. Assume that the alphabet always consists of the lower case letters 'a' to 'z'.

For instance, if the list only contains the combinations `q`, `ab` and `aaa`, the words would be enumerated like this:

1. a
2. b
⋮
16. p
17. r
⋮
26. aa
27. ac
⋮
649. zz
650. aac

Given the list of letter combinations, write a program that for a given word outputs its number, and for a given number ouputs its word. You can assume that none of the words will exceed 20 characters and no number will be greater than 2000000000 (for both input and output).

Input

The input will contain several test cases. The number of test cases T appears on a line by itself. Then follow T test cases. Each test case starts with a line containing two integers, N (the number of letter combinations, non-negative, at most 1000) and M (the number of queries for this list, positive, at most 100). Then follow N lines, each containing a lower case letter combination (between 1 and 3 letters, inclusive). After that follow M lines, each containing either a positive integer or a lower case word. If it's a word, it will not contain any of the combinations of letters in the list for this test case. If it's a number, it will not be greater than the number of words in the language.

Output

For each query, output a single line containing either the word's corresponding number, or the number's corresponding word.

input

```
2
3 4
q
ab
aaa
16
r
27
aac
7 2
a
b
c
d
ef
ghi
ijk
102345678
ksvfuw
```

output

```
p
17
ac
650
xexgun
39174383
```