Codeforces Round #742 (Div. 2)

A. Domino Disaster

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice has a grid with 2 rows and n columns. She fully covers the grid using n dominoes of size 1×2 — Alice may place them vertically or horizontally, and each cell should be covered by exactly one domino.

Now, she decided to show one row of the grid to Bob. Help Bob and figure out what the other row of the grid looks like!

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 5000$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 100$) — the width of the grid.

The second line of each test case contains a string s consisting of n characters, each of which is either `L`, `R`, `U`, or `D`, representing the left, right, top, or bottom half of a domino, respectively (see notes for better understanding). This string represents one of the rows of the grid.

Additional constraint on the input: each input corresponds to at least one valid tiling.

Output

For each test case, output one string — the other row of the grid, using the same format as the input string. If there are multiple answers, print any.

Example

input Copy

```
4
1
U
2
LR
5
LRDLR
6
UUUUUU
```

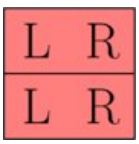
output Copy

```
D
LR
LRULR
DDDDDD
```

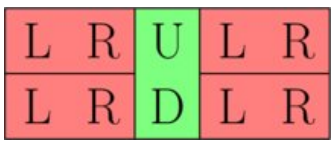
Note

In the first test case, Alice shows Bob the *top* row, the whole grid may look like:

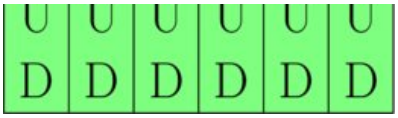
In the second test case, Alice shows Bob the *bottom* row, the whole grid may look like:



In the third test case, Alice shows Bob the *bottom* row, the whole grid may look like:



In the fourth test case, Alice shows Bob the *top* row, the whole grid may look like:



B. MEXor Mixup

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice gave Bob two integers a and b ($a > 0$ and $b \geq 0$). Being a curious boy, Bob wrote down an array of **non-negative** integers with MEX value of all elements equal to a and XOR value of all elements equal to b.

What is the *shortest* possible length of the array Bob wrote?

Recall that the MEX (Minimum EXcluded) of an array is the minimum non-negative integer that does **not** belong to the array and the XOR of an array is the bitwise XOR of all the elements of the array.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 5 \cdot 10^4$) — the number of test cases. The description of the test cases follows.

The only line of each test case contains two integers a and b ($1 \leq a \leq 3 \cdot 10^5$; $0 \leq b \leq 3 \cdot 10^5$) — the MEX and XOR of the array, respectively.

Output

For each test case, output one (positive) integer — the length of the shortest array with MEX a and XOR b. We can show that such an array always exists.

Example

input Copy

```
5
1 1
2 1
2 0
1 10000
2 10000
```

output Copy

```
3
2
3
2
3
```

Note

In the first test case, one of the shortest arrays with MEX 1 and XOR 1 is $[0, 2020, 2021]$.

In the second test case, one of the shortest arrays with MEX 2 and XOR 1 is $[0, 1]$.

It can be shown that these arrays are the shortest arrays possible.

C. Carrying Conundrum

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice has just learned addition. However, she hasn't learned the concept of "carrying" fully — instead of carrying to the *next* column, she carries to the column *two columns to the left*.

For example, the **regular** way to evaluate the sum $2039 + 2976$ would be as shown:

$$\begin{array}{cccccc} & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\ & 2 & 0 & 3 & 9 & \\ + & 2 & 9 & 7 & 6 & \\ \hline & 5 & 0 & 1 & 5 & \end{array}$$

However, Alice evaluates it as shown:

$$\begin{array}{cccccc} {\scriptstyle 1} & & {\scriptstyle 1} & {\scriptstyle 1} & & \\ & 2 & 0 & 3 & 9 & \\ + & 2 & 9 & 7 & 6 & \\ \hline \end{array}$$

$$\begin{array}{cccccc} & 2 & 9 & 7 & 6 \\ + & & & & \\ \hline 1 & 5 & 0 & 0 & 5 \end{array}$$

In particular, this is what she does:

- add 9 and 6 to make 15, and carry the 1 to the column *two columns to the left*, i. e. to the column "$0\ 9$";
- add 3 and 7 to make 10 and carry the 1 to the column *two columns to the left*, i. e. to the column "$2\ 2$";
- add 1, 0, and 9 to make 10 and carry the 1 to the column *two columns to the left*, i. e. to the column above the plus sign;
- add 1, 2 and 2 to make 5;
- add 1 to make 1.

Thus, she ends up with the incorrect result of 15005.

Alice comes up to Bob and says that she has added two numbers to get a result of n. However, Bob knows that Alice adds in her own way. Help Bob find the number of *ordered pairs of positive integers* such that when Alice adds them, she will get a result of n. Note that pairs (a, b) and (b, a) are considered different if $a \neq b$.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 1000$) — the number of test cases. The description of the test cases follows.

The only line of each test case contains an integer n ($2 \leq n \leq 10^9$) — the number Alice shows Bob.

Output

For each test case, output one integer — the number of *ordered* pairs of **positive integers** such that when Alice adds them, she will get a result of n.

Example

input

```
5
100
12
8
2021
10000
```

output

```
9
4
7
44
99
```

Note

In the first test case, when Alice evaluates any of the sums $1 + 9$, $2 + 8$, $3 + 7$, $4 + 6$, $5 + 5$, $6 + 4$, $7 + 3$, $8 + 2$, or $9 + 1$, she will get a result of 100. The picture below shows how Alice evaluates $6 + 4$:

$$\begin{array}{ccc} {\scriptstyle 1} & & \\ & & 6 \\ + & & 4 \\ \hline 1 & 0 & 0 \end{array}$$

D. Expression Evaluation Error

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

On the board, Bob wrote n positive integers in base 10 with sum s (i. e. in decimal numeral system). Alice sees the board, but accidentally interprets the numbers on the board as base-11 integers and adds them up (in base 11).

What numbers should Bob write on the board, so Alice's sum is as large as possible?

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases. The description of the test cases follows.

The only line of each test case contains two integers s and n ($1 \leq s \leq 10^9$; $1 \leq n \leq \min(100, s)$) — the sum and amount of numbers on the board, respectively. Numbers s and n are given in decimal notation (base 10).

Output

For each test case, output n positive integers — the numbers Bob should write on the board, so Alice's sum is as large as possible. If there are multiple answers, print any of them.

Example

input Copy

```
6
97 2
17 1
111 4
100 2
10 9
999999 3
```

output Copy

```
70 27
17
3 4 100 4
10 90
1 1 2 1 1 1 1 1 1
999900 90 9
```

Note

In the first test case, $70_{10} + 27_{10} = 97_{10}$, and Alice's sum is

$$70_{11} + 27_{11} = 97_{11} = 9 \cdot 11 + 7 = 106_{10}.$$

(Here x_b represents the number x in base b.) It can be shown that it is impossible for Alice to get a larger sum than 106_{10}.

In the second test case, Bob can only write a single number on the board, so he must write 17.

In the third test case, $3_{10} + 4_{10} + 100_{10} + 4_{10} = 111_{10}$, and Alice's sum is

$$3_{11} + 4_{11} + 100_{11} + 4_{11} = 110_{11} = 1 \cdot 11^2 + 1 \cdot 11 = 132_{10}.$$

It can be shown that it is impossible for Alice to get a larger sum than 132_{10}.

E. Non-Decreasing Dilemma

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice has recently received an array $a_1, a_2, \ldots, a_n$ for her birthday! She is very proud of her array, and when she showed her friend Bob the array, he was very happy with her present too!

However, soon Bob became curious, and as any sane friend would do, asked Alice to perform q operations of two types on her array:

- $1\ x\ y$: update the element a_x to y (set $a_x = y$).
- $2\ l\ r$: calculate how many non-decreasing subarrays exist within the subarray $[a_l, a_{l+1}, \ldots, a_r]$. More formally, count the number of pairs of integers (p, q) such that $l \leq p \leq q \leq r$ and $a_p \leq a_{p+1} \leq \cdots \leq a_{q-1} \leq a_q$.

Help Alice answer Bob's queries!

Input

The first line contains two integers n and q ($1 \leq n, q \leq 2 \cdot 10^5$) — the size of the array, and the number of queries, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the elements of Alice's array.

The next q lines consist of three integers each. The first integer of the i-th line is t_i, the operation being performed on the i-th step ($t_i = 1$ or $t_i = 2$).

If $t_i = 1$, the next two integers are x_i and y_i ($1 \leq x_i \leq n$; $1 \leq y_i \leq 10^9$), updating the element at position x_i to y_i (setting $a_{x_i} = y_i$).

If $t_i = 2$, the next two integers are l_i and r_i ($1 \leq l_i \leq r_i \leq n$), the two indices Bob asks Alice about for the i-th query.

It's guaranteed that there is at least one operation of the second type.

Output

For each query of type 2, print a single integer, the answer to the query.

Example

input Copy

```
5 6
3 1 4 1 5
2 2 5
2 1 3
1 4 4
2 2 5
```

```
1 2 6
2 2 5
```

output Copy

```
6
4
10
7
```

Note

For the first query, $l = 2$ and $r = 5$, and the non-decreasing subarrays $[p, q]$ are $[2,2]$, $[3,3]$, $[4,4]$, $[5,5]$, $[2,3]$ and $[4,5]$.

F. One-Four Overload

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice has an empty grid with n rows and m columns. Some of the cells are marked, and **no marked cells are adjacent to the edge of the grid**. (Two squares are *adjacent* if they share a side.)

Alice wants to fill each cell with a number such that the following statements are true:

- every *unmarked* cell contains either the number 1 or 4;
- every *marked* cell contains the sum of the numbers in all **unmarked** cells adjacent to it (if a marked cell is not adjacent to any unmarked cell, this sum is 0);
- every *marked* cell contains a multiple of 5.

Alice couldn't figure it out, so she asks Bob to help her. Help Bob find any such grid, or state that no such grid exists.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 500$) — the number of rows and the number of columns in the grid, respectively.

Then n lines follow, each containing m characters. Each of these characters is either '`.`' or '`X`' — an unmarked and a marked cell, respectively. **No marked cells are adjacent to the edge of the grid.**

Output

Output "`NO`" if no suitable grid exists. Otherwise, output "`YES`". Then output n lines of m space-separated integers — the integers in the grid.

Examples

input Copy

```
5 5
.....
.XXX.
.X.X.
.XXX.
.....
```

output Copy

```
YES
4 1 4 4 1
4 5 5 5 1
4 5 1 5 4
1 5 5 5 4
1 4 4 1 4
```

input Copy

```
5 5
.....
.XXX.
.XXX.
.XXX.
.....
```

output Copy

```
NO
```

input Copy

```
3 2
..
..
..
```

output Copy

```
YES
```

```
4 1
4 1
1 4
```

input Copy

```
9 9
.........
.XXXXX.X.
.X...X...
.X.XXXXX.
.X.X.X.X.
.X.XXX.X.
.X.....X.
.XXXXXXX.
.........
```

output Copy

```
YES
4 4 4 1 4 1 4 1 4
1 5 5 5 5 5 4 10 1
4 5 1 4 1 5 4 4 4
4 5 1 5 5 0 5 5 1
4 5 1 5 4 5 1 5 4
4 5 1 5 5 5 4 5 1
1 5 4 4 1 1 4 5 1
4 5 5 5 5 5 5 5 4
1 1 1 1 4 4 1 1 4
```

Note

It can be shown that no such grid exists for the second test.