Intra-BUET Programming Contest 2026

https://toph.co/contests/training/kalcqb8

Toph

Schedule

The contest will run for **4h0m0s**.

Authors

The authors of this contest are A_Nafees, ashraful7525, kotoshogiku.510125, Shafin, Tahjib31, Tahmid690, Waseem_Mustak, and Zeeman01.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use C++17 GCC 13.2, C++20 Clang 16.0, C++20 GCC 13.2, C++23 GCC 13.2, C11 GCC 13.2, C17 GCC 13.2, C23 GCC 13.2, Java 1.8, Kotlin 1.9, Kotlin 2.0, PyPy 7.3 (3.10), and Python 3.12 in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 10 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

Disclaimer

The contents of this contest have not been reviewed by Toph and do not necessarily represent Toph's views.

A. Biye!!!

Congratulations! After years of struggle, you are finally getting married. The venue is set, the guests have arrived, and the aroma of Kacchi Biryani fills the air. However, a crisis has emerged that threatens the prestige ("Loke ki bolbe") of your family.

You are serving dinner to N VIP guests seated in a single row at the Head Table. Currently, the size of the mutton piece on the i-th guest's plate is denoted by A_i.

Unfortunately, your Father-in-Law has invited the dreaded Judgmental Auntie Squad. These Aunties do not look at plates individually; they scan the table in pairs to find flaws. For any two adjacent guests at indices i and i + 1, the Aunties will strictly judge that section based on the largest piece of meat between them. The smaller piece is ignored, but the larger piece sets the "status" for that pair.

Your Father-in-Law has provided a "Vision of Perfection" represented by an array B. To avoid a scandal, the Aunties' judgment of your table must perfectly match their judgment of the Father-in-Law's vision.

Formally, for every adjacent pair of guests $(1 \leq i < N)$, the following condition must hold:

$$max(A'_i, A'_{i+1}) = max(B_i, B_{i+1})$$

Where A' is the modified state of your array A.

You are currently hiding under the table with a toolkit. In one operation, you can:

- Increment the size of the mutton piece on a specific plate by 1 (using edible glue).
- Decrement the size of the mutton piece on a specific plate by 1 (using a knife).

Your task is to determine the minimum total number of operations required to satisfy the Aunties and save your marriage.

Input

The first line contains a single integer $N (2 \leq N \leq 2 \times 10^5)$, the number of guests.

The second line contains N space-separated integers $A_1, A_2, ..., A_N (1 \leq A_i \leq 10^9)$, the initial sizes of the mutton pieces.

The third line contains N space-separated integers $B_1, B_2, ..., B_N (1 \leq B_i \leq 10^9)$, the reference sizes from the Father-in-Law's vision.

Output

Print a single integer representing the minimum number of operations required to satisfy the condition.

Examples

Input	Output
3 2 5 2 2 3 2	2

- Current Table (A): [2, 5, 2]. The Aunties see maximums of max(2, 5) = 5 and max(5, 2) = 5.
- Vision (B): [2, 3, 2]. The Aunties expect maximums of max(2, 3) = 3 and max(3, 2) = 3.
- The Fix: You act quickly to trim the middle piece (A2) from 5 down to 3.
- Result: New A′ = [2, 3, 2]. The pair maximums are now 3 and 3. The Aunties are satisfied.
- Total cost = 2 operations

Input	Output
4 4 4 4 4 1 2 3 4	5

B. The Case of the Silent Vaults

> Data! Data! Data! I can't make bricks without clay - *Sherlock Holmes, The Adventure of the Copper Beeches*

London, 1895. A thick fog envelops the city, but a graver matter clouds the mind of Sherlock Holmes. Professor Moriarty has infiltrated the subterranean levels of the Bank of England. He has not stolen the gold directly; rather, he has tampered with the distribution of sovereigns across the vaults to create chaos in the British economy.

The Bank possesses a long corridor of N vaults, indexed sequentially from 1 to N. Initially, the i-th vault contains a_i gold sovereigns.

Moriarty, strictly adhering to a mathematical pattern, sends his henchmen to modify the contents of the vaults in specific ranges. Holmes, however, has deduced that the key to reversing the damage lies not in checking every single vault, but in summing the contents of vaults that align with specific harmonic frequencies (multiples of an integer k).

As Holmes's trusted assistant, you must maintain the ledger of the vaults and answer Holmes's queries rapidly before the bank opens in the morning.

You must process Q events, which occur in chronological order. There are two types of events:

- Moriarty's Tampering - $1\ l\ r\ x$

  Moriarty's henchmen add x sovereigns to every vault from index l to r inclusive.

  - For all $l \le i \le r$:

    $$a_i = a_i + x \quad \text{for all } \ l \le i \le r$$

  - *Note: If x is negative, they are removing gold.*

- Holmes's Deduction - $2\ l\ r\ k$

  Holmes requires the total number of sovereigns stored in vaults within the range $[l, r]$ whose index is perfectly divisible by k.

  - Compute and output:

$$\sum_{i=l}^{r} a_i \quad \text{where} \quad i \equiv 0 \pmod{k}$$

Input

The first line contains a single integer T $(1 \leq T \leq 10)$ — the number of test cases.

For each test case the first line contains two integers N and Q ($1 \leq N \leq 3 \times 10^5$, $1 \leq Q \leq 5 \times 10^5$) — the number of vaults and the number of events.

The second line contains N integers, where the i-th integer denotes the initial gold a_i ($-10^6 \leq a_i \leq 10^6$).

The following Q lines each describe an event in the formats:

- $1\ l\ r\ x$ ($1 \leq l \leq r \leq N, -10^6 \leq x \leq 10^6$)
- $2\ l\ r\ k$ ($1 \leq l \leq r \leq N, 1 \leq k \leq N$)

It is guaranteed that the sum of N over all test cases does not exceed $3 \cdot 10^5$ and sum of Q does not exceed $5 \cdot 10^5$

Output

For every Type 2 event (Holmes's Deduction), output the calculated sum on a single line.
Note : Use faster input output method

Example

<table>
<tr><td><u>Input</u></td><td><u>Output</u></td></tr>
<tr><td><pre>1
10 5
1 2 3 4 5 6 7 8 9 10
2 1 10 2
1 1 5 2
2 1 10 2
2 1 10 3
2 5 9 1</pre></td><td><pre>30
34
20
37</pre></td></tr>
<tr><td colspan="2">1. Query (1 10 2): Holmes checks multiples of 2: indices {2, 4, 6, 8, 10}. Sum: 2 + 4 + 6 + 8 + 10 = 30.

2. Update (1 5 2): Moriarty adds 2 sovereigns to vaults 1 through 5. New state: [3, 4, 5, 6, 7, 6, 7, 8, 9, 10].</td></tr>
</table>

Input	Output
3. Query (1 10 2): Indices {2, 4, 6, 8, 10} now hold {4, 6, 6, 8, 10}. Sum: 34. 4. Query (1 10 3): Holmes checks multiples of 3: indices {3, 6, 9}. Values are now {5, 6, 9}. Sum: 20. 5. Query (5 9 1): Holmes checks multiples of 1: Sum: 37.	

C. Connecting Everything

Let's first define a function F that takes in an array arr of length n, and outputs a directed graph G which has n nodes and n edges, such that, for all $1 \le i \le n$, there is a directed edge from arr_i to i.

For example, $F([1,1,3,2,5])$ generates a directed graph which has the following edges: $(1 \to 1)$, $(1 \to 2)$, $(3 \to 3)$, $(2 \to 4)$, $(5 \to 5)$.

A graph G is said to be beautiful if the graph forms a single strongly connected component (starting from any of the nodes, you can reach any other node following a path in the graph).

The graph generated above is not beautiful because you can not reach 2, if you start from 4.

But the graph generated by $F([5,1,2,3,4])$ is beautiful.

For this problem, you will be given a permutation P of length n.

> A **permutation** of length n is an array consisting of n distinct integers from 1 to n in any order. For example, [2, 3, 1, 5, 4] is a permutation, but [1, 2, 2] is not (it contains duplicate elements) and [1, 3, 4] is not (it is missing the number 2).

In one operation, you can choose any $1 \le i \le n$ and $1 \le x \le n$, and set $P_i = x$. Please output the minimum number of operations to make $F(P)$ beautiful. Also output the operations themselves.

Please note that, after doing any operation, P does not need to be a permutation.

Input

The first line contains an integer $t (1 \le t \le 10^5)$, the number of test cases.

In the first line of each test case, an integer n is given --- the length of the permutation $(1 \le n \le 10^5)$.

In the second line of each test case, n integers P_i are given --- the elements of the permutation $(1 \le P_i \le n)$. It is guaranteed that all P_i are distinct.

It is guaranteed that the sum of P_i across all test cases does not exceed 10^5.

Output

For each test case, in a single line, first output k, the number of operations required.

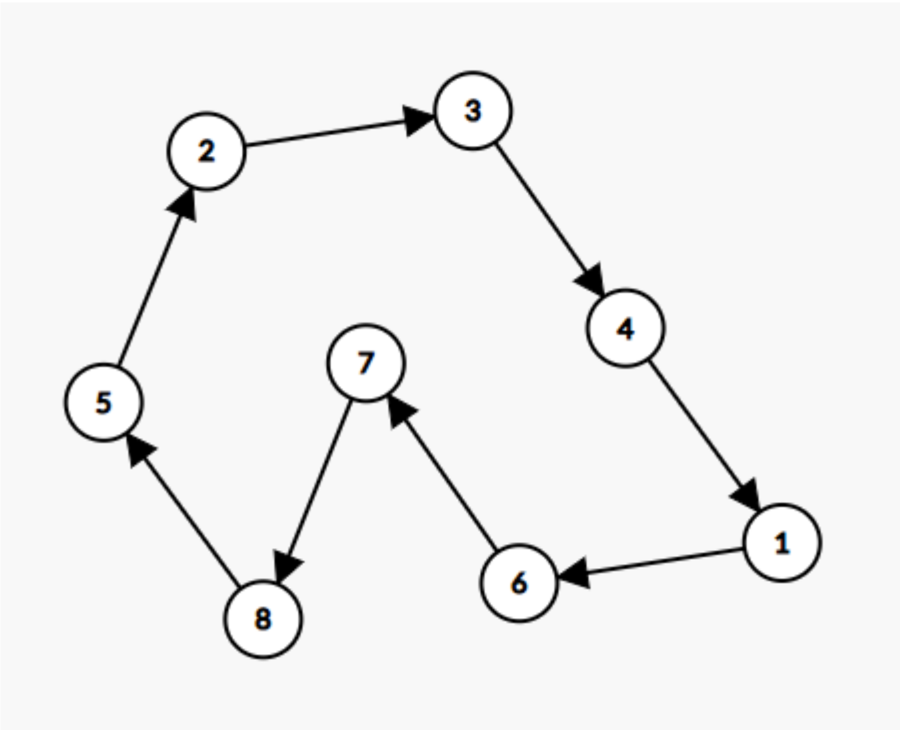
Then, output k lines. In the j-th line, print two integers : i_j and x_j, which denotes the j-th operation

Example

<u>Input</u>	<u>Output</u>
4 5 5 1 2 3 4 8 4 1 2 3 8 5 6 7 5 1 2 3 4 5 9 4 1 2 3 8 5 6 7 9	0 2 6 1 2 5 5 2 1 3 2 4 3 5 4 1 5 3 6 1 9 5 2 9

In the first test case, the generated graph already forms a strongly connected component, so, no operations are necessary.

In the second case, the generated graph looks like this, which is a beautiful graph.



D. Island Connectivity

In the year 2126, the archipelago of n islands plans to modernize its transportation system. There are m proposals for building teleporters. The i-th proposal involves a connection between island u_i and island v_i, generating a potential profit of w_i if constructed.

However, the current generation of teleporter technology has specific technical limitations:

- If a teleporter is constructed between islands u and v, it must be configured to function in strictly **one direction** (either u to v or v to u, but not both).
- Each island possesses limited power reserves and can support **at most one** outgoing teleporter connection. (Note: There is no limit on the number of incoming teleporters an island can receive).

As the Chief Planning Officer, you must select a subset of proposals and assign a direction to each selected teleporter such that all technical constraints are met and the total profit is maximized.

Calculate the maximum possible total profit.

Input

There are T $(1 \leq T \leq 10^4)$ independent test cases. The first line consists of a single integer T. The description of T test cases follows.

The first line of each test case contains two integers n and m ($2 \leq n \leq 2 \times 10^5$, $1 \leq m \leq 2 \times 10^5$) — the number of islands and the number of proposed teleporters.

The next m lines of each test case contain three integers u_i, v_i, and w_i ($1 \leq u_i, v_i \leq n, u_i != v_i$, $1 \leq w_i \leq 10^9$) — representing a proposal to connect island u_i and v_i with profit w_i.

It is guaranteed that there cannot be multiple proposals between the same pair of islands.

The sum of n and the sum of m over all the test cases do not exceed 2×10^5.

Output

For each test case, print a single integer on a single line. — the maximum total profit achievable while satisfying all constraints.

Example

Input	Output
2 4 5 1 2 10 2 3 10 3 4 10 4 1 20 2 4 10 3 2 1 3 20 2 3 30	50 50

E. The Beautiful Reunion

Yunus, Trump, and Ronaldo live on an **acute triangular island** defined by three vertices A, B, and C.

Local legends speak of **Beautiful Points** located strictly inside the island. A point X is defined as *beautiful* if and only if the circumradii of the three triangles formed by connecting it to the island's vertices are equal:

$$\text{Radius}(\triangle XAB) = \text{Radius}(\triangle XBC) = \text{Radius}(\triangle XCA)$$

One day, the three friends meet at the **Circumcenter** of the island (the center of the unique circle passing through A, B, and C). They decide to separate and reunite at a Beautiful Point X to celebrate. However, they must follow a specific ritual:

- **Yunus** travels in a straight line from the Circumcenter to a point P_{AB} on edge AB, then travels in a straight line from P_{AB} to X.
- **Trump** travels in a straight line from the Circumcenter to a point P_{BC} on edge BC, then travels in a straight line from P_{BC} to X.
- **Ronaldo** travels in a straight line from the Circumcenter to a point P_{CA} on edge CA, then travels in a straight line from P_{CA} to X.

All three friends move at the **same constant speed**. They must select a Beautiful Point X and their respective touch-points (P_{AB}, P_{BC}, P_{CA}) such that they all arrive at X at the **exact same time**.

Your task is to find the coordinates of X and the touch-points. If there are multiple valid Beautiful Points or multiple valid paths, choose the solution that requires the **minimum time** for the reunion.

Input

The first line contains a single integer $t (1 \leq t \leq 10^3)$ — the number of test cases.

Each test case consists of a single line containing 6 integers $x_A, y_A, x_B, y_B, x_C, y_C (-10^9 \leq x, y \leq 10^9)$.

It is guaranteed that the points A, B, C form a valid acute triangle.

Output

For each test case:

- If no such Beautiful Point X exists strictly inside the triangle, print -1.
- Otherwise, print 8 numbers in a single line separated by spaces:
  $X_x \quad X_y \quad P_{AB_x} \quad P_{AB_y} \quad P_{BC_x} \quad P_{BC_y} \quad P_{CA_x} \quad P_{CA_y}$

Where (X_x, X_y) are the coordinates of the chosen Beautiful Point, and the subsequent pairs are the coordinates of the touch-points for Yunus (on AB), Trump (on BC), and Ronaldo (on CA).

Your answer will be considered correct if the absolute or relative error of each coordinate does not exceed 10^{-6}.

Example

Input	Output
3 2 1 14 1 9 8 10 10 30 10 20 40 9 1 1 1 5 11	9 6 8.166666666667 1 10.276595744681 6.212765957447 7.285714285714 6.285714285714 20 13.333333333333 20 10 26.153846153846 21.538461538462 13.846153846154 21.538461538462 5 2.6 5 1 2.633802816901 5.084507042254 7.366197183099 5.084507042254

Use *long double* in *C/C++* to avoid precision error.

F. Operation: Null Divergence

The mad scientist **Hououin Kyouma** (aka Okabe Rintarou) has discovered a terrifying truth: the current World Line is unstable. To reach the **Steins;Gate**—the only timeline where Mayuri and Kurisu are both safe—the Divergence Meter must read exactly `0000000`.

However, the Organization has encrypted the Divergence Meter's display into a binary sequence, S. To shift the World Line, Lab member 003, **Daru**, has hacked into the SERN database to perform a "Micro-Leap" operation.

A "Micro-Leap" consists of choosing any two **adjacent** bits in the binary sequence and **inverting both of them** (changing `0` to `1` or `1` to `0`).

Okabe asks you: Given the initial binary string S, is it possible to make the entire sequence consist of only `0`s by performing the Micro-Leap operation any number of times?

If the string can be converted to all zeros, the Steins;Gate world line is reachable.

Input

The first line contains an integer T ($1 \le T \le 100$) — the number of test cases.

For each test case, the first line contains a binary string S ($1 \le |S| \le 10^5$) consisting of characters `0` and `1`

Output

For each test case, print `Yes` if it is possible to make the string all zeros, and `No` otherwise.

Example

Input	Output
3 0110 111 1001	Yes No Yes

<table>
<tr><th><u>Input</u></th><th><u>Output</u></th></tr>
<tr><td colspan="2">Explanation for Test Case 3 (1001):

1. Invert indices 2 and 3: <code>1001</code> → <code>1111</code>
2. Invert indices 1 and 2: <code>1111</code> → <code>0011</code>
3. Invert indices 3 and 4: <code>0011</code> → <code>0000</code>

Result: Yes.</td></tr>
</table>

G. The Real Folk Blues I

"I'm not going there to die. I'm going to find out if I'm really alive." --- Spike Spiegel

This is the **easy version** of the problem. In this version, you can make at most **160 queries.**

Vicious has gone underground. The Red Dragon Syndicate is protecting his location, encrypted behind a complex mathematical signal. Spike Spiegel has intercepted the Syndicate's communication line. However, the data is heavily obscured. Spike cannot read Vicious's coordinates directly. He can only ping the network with an offset, and the system returns the "radical signature" of the location.

Spike knows Vicious is hiding at a coordinate n $(1 \leq n \leq 2 \times 10^5)$ To find him, Spike can choose a non-negative integer x $(0 \leq x \leq 2 \times 10^5)$ and hack the system with a query. The Syndicate's server will reply with the **set of prime divisors** of the number $n + x$.

For example, if Vicious is at $n = 12$ and Spike hacks the system with offset $x = 16$, the system replies with $\{2, 7\}$ because $16 + 12 = 28 = 2^2 \times 7$.

Spike needs to pinpoint the exact value of n to settle the score. But he cannot make more than **160 queries** — otherwise, Vicious will detect Spike's presence and kill him.

Can you help Spike locate Vicious?

Interaction Protocol

To make a query, print a line in the format: `? x`

After printing a query, do not forget to output the end of line and flush the output.

After each query, read an integer k, the number of distinct prime divisor of $n + x$ followed by the k prime divisors $p_1, p_2, \ldots p_k$.

When you have determined the hidden number n, print the answer in the format: `! n` This output does not count towards the query limit. After printing the answer, your program must terminate immediately.

Input

The input consists of the responses to your queries. After each query `? x`, you will recieve:

- an integer k: the number of distinct prime divisor of $n + x$
- followed by k space separated integer : the distinct prime factors of $n + x$

Output

To ask a query, output "`? x`" (without quotes), where $0 \le x \le 2 \times 10^5$, followed by a flush operation. When you determine the value of n, output "`! n`" (without quotes), where $1 \le n \le 2 \times 10^5$ and flush the output.

To flush the output buffer, use:

- `fflush(stdout)` or `cout << endl` in C++.
- `System.out.flush()` in Java.
- `sys.stdout.flush()` in Python.

Example Interaction

Input	Output
	? 0
3 2 3 5	
	? 7
1 37	
	! 30

In the example above, the hidden number is $n = 30$.

- The first query is "`? 0`". The system calculates $30 + 0 = 30$. The prime factors of 30 are 2, 3, and 5. The input receives the count `3` followed by the primes `2 3 5`.
- The second query is "`? 7`". The system calculates $30 + 7 = 37$, which is prime. The input receives count `1` followed by the prime `37`.
- The user deduces $n = 30$ and prints "`! 30`".

H. The Real Folk Blues II

"You're gonna carry that weight'' --- From Cowboy Bebop

This is the **hard version** of the problem. In this version, you can make at most **2 queries.**

Vicious has gone underground. The Red Dragon Syndicate is protecting his location, encrypted behind a complex mathematical signal. Spike Spiegel has intercepted the Syndicate's communication line. However, the data is heavily obscured. Spike cannot read Vicious's coordinates directly. He can only ping the network with an offset, and the system returns the "radical signature" of the location.

Spike knows Vicious is hiding at a coordinate n $(1 \le n \le 2 \times 10^5)$ To find him, Spike can choose a non-negative integer x $(0 \le x \le 2 \times 10^5)$ and hack the system with a query. The Syndicate's server will reply with the **set of prime divisors** of the number $n + x$.

For example, if Vicious is at $n = 12$ and Spike hacks the system with offset $x = 16$, the system replies with $\{2, 7\}$ because $12 + 16 = 28 = 2^2 \times 7$.

Spike needs to pinpoint the exact value of n to settle the score. But he cannot make more than **2 queries** — otherwise, Vicious will detect Spike's presence and kill him.

Can you help Spike locate Vicious?

Interaction Protocol

To make a query, print a line in the format: `? x`

After printing a query, do not forget to output the end of line and flush the output.

After each query, read an integer k, the number of distinct prime divisor of $n + x$ followed by the k prime divisors $p_1, p_2, \ldots p_k$.

When you have determined the hidden number n, print the answer in the format: `! n` This output does not count towards the query limit. After printing the answer, your program must terminate immediately.

Input

The input consists of the responses to your queries. After each query `? x`, you will recieve:

- an integer k: the number of distinct prime divisor of $n + x$
- followed by k space separated integer : the distinct prime factors of $n + x$

Output

To ask a query, output "`? x`" (without quotes), where $0 \leq x \leq 2 \times 10^5$, followed by a flush operation.

When you determine the value of n, output "`! n`" (without quotes), where $1 \leq n \leq 2 \times 10^5$ and flush the output.

To flush the output buffer, use:

- `fflush(stdout)` or `cout << endl` in C++.
- `System.out.flush()` in Java.
- `sys.stdout.flush()` in Python.

Example Interaction

Input	Output
	? 0
3 2 3 5	
	? 7
1 37	
	! 30

In the example above, the hidden number is $n = 30$.

- The first query is "`? 0`". The system calculates $30 + 0 = 30$. The prime factors of 30 are 2, 3, and 5. The input receives the count `3` followed by the primes `2 3 5`.
- The second query is "`? 7`". The system calculates $30 + 7 = 37$, which is prime. The input receives count `1` followed by the prime `37`.
- The user deduces $n = 30$ and prints "`! 30`".

I. The Question Bank and the thief

Tahmid and Shafin, members of the **Hotasha Gang**, have a question bank. They plan to use some of these problems in the upcoming **Intra BUET Programming Contest**, so they secure the question bank with a password. Tanvir, a participant in the contest, is worried about the difficulty of the problems and decides to steal some questions from the bank. Unfortunately, he does not know the password. However, Tanvir overhears Shafin telling Tahmid,

“The password is a 4-digit number that is generated from another 4-digit number N using the following process: We calculate the sum of all possible numbers formed from every **contiguous subsegment** of N. This sum is the password.''

Tanvir wants to determine whether a given 4-digit number can generate a valid password. Formally, you'll be given a 4-digit number N. If the calculated sum is a **4-digit number**, output “`Yes`”, otherwise output “`No`”.

A **contiguous subsegment** is any substring formed by consecutive digits of the number. For example, if $N = 1222$, the subsegments are: $1, 2, 2, 2, 12, 22, 22, 122, 222, 1222$

Leading zeros are allowed while forming subsegments (for example, the subsegment “02” is considered as the number 2).

Input

- The first line contains an integer T $(1 \leq T \leq 5000)$, the number of test cases.
- Each of the next T lines contains a single 4-digit integer N. More formally $1000 \leq N \leq 9999$.

Output

- For each test case, print “`Yes`" or “`No`" on a separate line.

Example

Input	Output
3 1222	Yes No Yes

<u>Input</u>	<u>Output</u>
`9999` `1010`	

- For the first test case, the sum of all the continuous subsegments of 1222 is $1+2+2+2+12+22+22+122+222+1222=1629$, is a 4-digit number. So 1222 can generate valid password.
- For the second test case, the sum of all the continuous subsegments of 9999 is $9+9+9+9+99+99+99+999+999+9999=12330$, is a 5-digit number. So 9999 can't generate valid password.
- For the third test case, the sum of all the continuous subsegments of 1010 is $1+0+1+0+10+01+10+101+010+1010=1144$, is a 4-digit number. So 1010 can generate valid password.

J. String Reduction

Tahjib has a binary string (string consisting only of '0' and '1') S of length n. From this original string, he generates a collection of n new strings. To create the i-th string in this collection, he removes exactly the character at the i-th index of S, keeping the relative order of the remaining characters intact.

He tasks you with organizing these generated strings of length $n-1$. Specifically, Tahjib asks you to determine which string would appear at the k-th position if all n generated strings were sorted in the lexicographical order.

See sample explanation below for more clarity.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) --- the number of test cases.

The first line of each test case contains two integers n and k ($2 \le n \le 2 \times 10^5$, $1 \le k \le n$) --- the length of the original string and the target position.

The second line of each test case contains the binary string S of length n.

The sum of n over all test cases does not exceed 2×10^5.

Output

For each test case, output a single line containing the k-th lexicographically smallest string among the n generated strings.

Example

Input	Output
3 4 2 1010 3 1 111 5 5 11011	100 11 1111

In the first test case:

The string is 1010. The generated strings are:

1. Remove index 1: `010`
2. Remove index 2: `110`
3. Remove index 3: `100`
4. Remove index 4: `101`

Sorting these gives: ["010", "100", "101", "110"].

Since $k = 2$, the answer is the 2nd string: 100.

In the second test case:

The string is 111. Removing any character results in 11.

The list is ["11", "11", "11"]. The 1st string is 11.