The International Collegiate Programming Contest
Sponsored by ICPC Foundation

The 2025 Teens online Qualifications Collegiate Programming Contest

(Contest Problems)

Egypt
April 2025

Problem A. ramen and his love

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red`

Ramen has a secret love, but he is trying to hide her name. His friend, Sohil, claims to have figured out the secret and teases Ramen about it. To prove his knowledge, Ramen asks Sohil to solve two types of queries about the string that represents the secret name.

You are given a string S of lowercase English letters. Sohil needs to answer the following two types of queries:

- **Type 1:** Given two integers l and r ($1 \le l \le r \le n$), determine how many unique characters exist in the substring $S[l : r]$.
- **Type 2:** Given a position p ($1 \le p \le n$) and a character c, replace the character at position p in the string S with c.

Ramen will ask these queries q times, and you are tasked with helping Sohil answer them efficiently.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$) — the length of the string and the number of queries.

The second line contains a string S of length n, consisting of lowercase English letters.

Each of the next q lines contains a query in one of the two formats:

- `1 l r` — a query of Type 1.
- `2 p c` — a query of Type 2.

Note: The string is 1-indexed.

Output

For each query of Type 1, output a single integer — the number of distinct characters in the specified range.

Example

standard input	standard output
12 5 icpcmansoura 1 1 6 1 4 11 2 6 x 1 1 6 1 4 11	5 8 5 8

Note

Initial string: `icpcmansoura`

- First query (1 1 6): Count unique characters in substring `icpcma` → `i, c, p, m, a` → Output: 5

- Second query (1 4 11): Count unique characters in substring `cmansour` → `c, m, a, n, s, o, u, r` → Output: 8
- Third query (2 6 x): Update the character at position 6 from `a` to `x`. The string becomes: `icpcmxnsoura`
- Fourth query (1 1 6): Count unique characters in substring `icpcmx` → `i, c, p, m, x` → Output: 5
- Fifth query (1 4 11): Count unique characters in substring `cmxnsour` → `c, m, x, n, s, o, u, r` → Output: 8

Problem B. Omani's Wings of Destiny

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Gray`

Omani is planning a long journey across N cities, numbered from 1 to N. He will start at city 1 and his goal is to reach city N while spending as little money as possible.

Omani has two ways to move between cities:

Bus Travel:

- He can travel from city i to city $i+1$ using a bus, which costs $A[i]$ units of money.

Flight Travel:

- He can take a flight from city i to any city j $(i < j)$ as long as $j - i \leq K$. The cost of this flight is $B[i] + B[j]$.
- However, Omani can take at most M flights during his entire journey.

Your task is to help Omani find the minimum total cost required to reach city N from city 1 under these rules.

Input

The first line contains three integers N, M, and K $(1 \leq N \leq 10^6, 0 \leq M \leq 10^{12}, 1 \leq K \leq N)$ — the number of cities, the maximum number of flights Omani can take, and the maximum flight distance (in terms of consecutive cities).

The second line contains $N-1$ integers $A[1], A[2], \ldots, A[N-1]$ $(1 \leq A[i] \leq 10^9)$, where $A[i]$ is the cost to travel from city i to city $i+1$ by bus.

The third line contains N integers $B[1], B[2], \ldots, B[N]$ $(1 \leq B[i] \leq 10^9)$, where $B[i]$ is the flight cost component for city i.

Output

Print a single integer — the minimum total cost required for Omani to travel from city 1 to city N.

Example

standard input	standard output
5 3 2 1 2 3 4 1 1 2 1 7	7

Problem C. Lucky Adham

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Pink`

You are given a string number n of length S and an integer k. Your task is to remove at most k digits from n to get the minimum possible value of n.

You are given a string n consisting of S digits. Adham considers a number *lucky* if it is as small as possible.

To make n lucky, you can remove up to k digits from it. The digits you remove can be chosen arbitrarily, but the order of the remaining digits must stay the same. Your task is to determine the smallest number you can obtain after removing at most k digits.

Input

The first line contains two integers S and k $(0 \leq k < S \leq 10^5)$ — the length of the number and the maximum number of digits you are allowed to remove.

The second line contains a string n of length S, consisting only of digits from 0 to 9.

Output

Print the smallest possible number obtained after removing at most k digits. The result should not contain leading zeros unless it is exactly 0.

Examples

standard input	standard output
6 3 123456	123
8 3 98475201	45201

Note

In the first example:

- The original number is `123456`.
- By removing the last three digits, the smallest possible number is `123`.

In the second example:

- The original number is `98475201`.
- The smallest result can be achieved by removing the digits `9`, `8`, and `7`, leaving `45201`.

Problem D. Mathematical roads

Input file:	standard input
Output file:	standard output
Balloon Color:	White

Sa7afy is playing a game about road architecture.

There are n cities and m one-way roads. Each road connects exactly two cities. You can use any road from city x to city y for free, any number of times, if it exists in the given direction.

Additionally, Sa7afy has a magic card that allows him to **flip** any road. Flipping a road means you can use it in the reverse direction. However, flipping is not free: **Each flip costs coins equal to the sum of all divisors of** k, where:

$$k = x + y$$

Here, x and y are the endpoints of the flipped road (x is the original start, y is the original end).

Sa7afy starts at city s and wants to reach city f. You need to determine the minimum total cost required to reach city f using the available roads and flips.

If it is impossible to reach city f, output -1.

Input

The first line contains four integers n, m, s, and f ($1 \le n \le 2 \cdot 10^5, 1 \le m \le 10^6, 1 \le s, f \le n$) — the number of cities, number of roads, starting city, and destination city.

Each of the next m lines contains two integers x and y ($1 \le x, y \le n$) — indicating a one-way road from city x to city y.

Output

Print a single integer — the minimum total cost to reach city f. If it is impossible to reach city f, print -1.

Example

standard input	standard output
5 4 1 5 1 2 2 3 3 4 5 4	13

Note

In the example:

- Sa7afy can flip the road from city 4 to city 5, so it becomes usable in reverse (5 to 4).
- The cost of this flip is the sum of divisors of $k = 4 + 5 = 9$, which is:

$$1 + 3 + 9 = 13$$

- He can then reach city 5 using this flipped road.

Problem E. AND Hagry Array

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Green`

Given an array a of length n, define the *score* of a subsequence c (of length m) as:

$$c_1 \,\&\, c_2 \,\&\, \ldots \,\&\, c_m$$

where & denotes the bitwise AND operation.

Your task is to determine the maximum possible length of a subsequence that achieves the maximum possible score.

Input

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

Each test case consists of two lines:

- The first line contains an integer n ($1 \le n \le 10^5$) — the length of the array a.
- The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — the elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the maximum length of a subsequence that achieves the maximum possible score.

Example

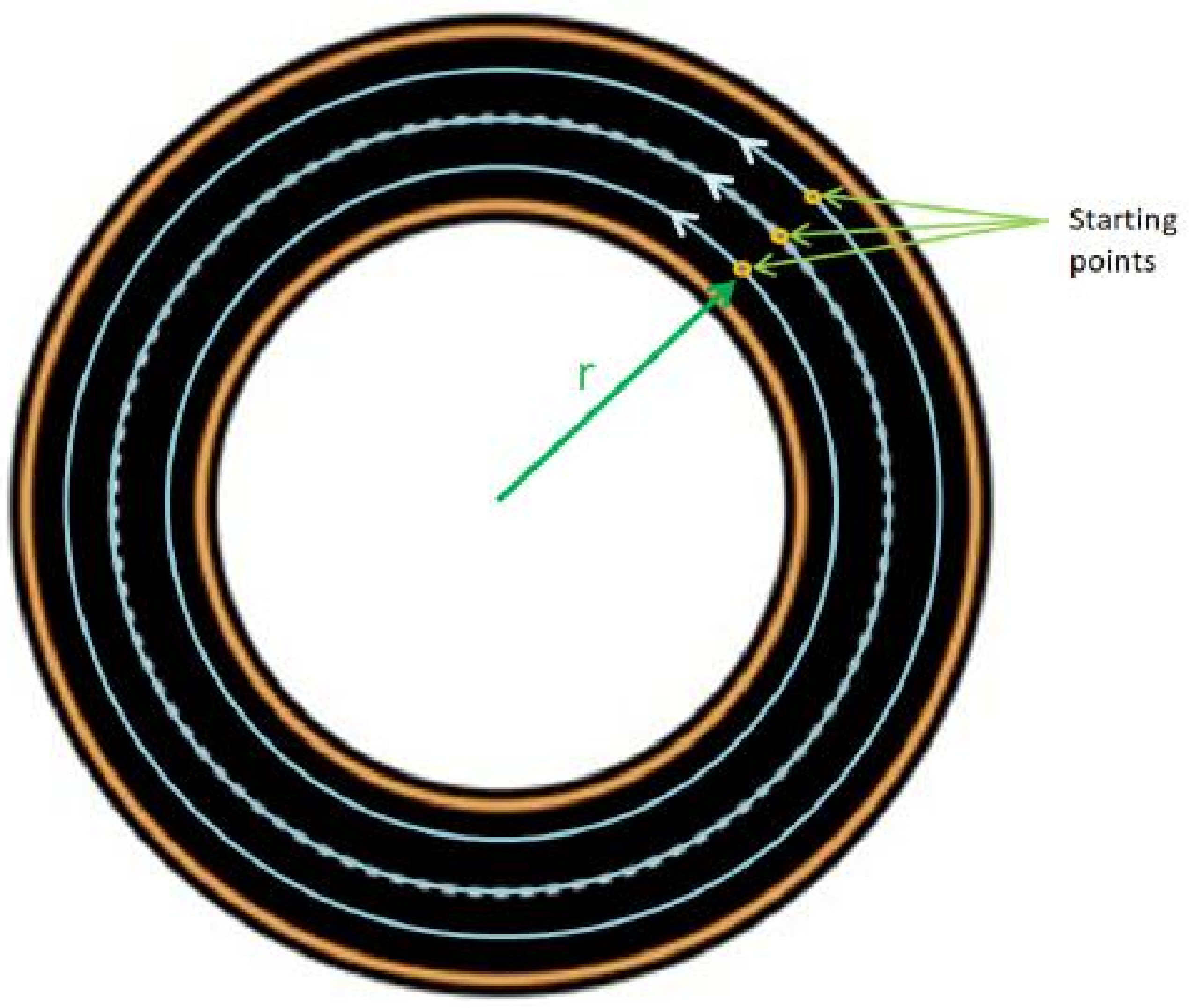
standard input	standard output
2 5 1 4 2 4 3 3 8 9 3	2 1

Problem F. Around The Road

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Black`

After Ramadan and Eid, you decided to lose weight by walking. Your friend suggested a daily routine: walk exactly n meters every day. To do this, you will walk along a circular road following these rules:

- The circular road has a center, and you start at a distance of r meters from the center.
- You walk along the circumference of the circle. Once you complete one full round, you shift to a new circle whose radius is increased by m meters.
- You must not walk the same path more than once, so each round must be on a new circle.
- When you finish walking the total of n meters, you may stop in the middle of a round.



Your task is to calculate your final distance from the center (i.e., the radius you have reached) after walking n meters.

Use π, defined as $\pi = \arccos(-1)$, for calculations.

Input

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

Each test case consists of one line containing three integers n, m, and r:

- n ($1 \le n \le 10^9$) — the total distance to walk.
- m ($1 \le m \le 10^4$) — the amount by which the radius increases after each completed round.
- r ($1 \le r \le 10^4$) — the initial distance from the center.

Output

For each test case, print a single integer — the radius of the circle you will be at after finishing the n meters walk.

Example

standard input	standard output
2 50 3 7 131 2 5	10 9

Problem G. Roman to Integer

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

Roman numerals use the following symbols:

$$I, V, X, L, C, D, M$$

Their values are:

- $I = 1$, $V = 5$, $X = 10$, $L = 50$, $C = 100$, $D = 500$, $M = 1000$

Normally, numerals are written from largest to smallest from left to right. However, to avoid repeating the same symbol more than three times, subtraction is used in specific cases:

- I before V or X makes 4 and 9.
- X before L or C makes 40 and 90.
- C before D or M makes 400 and 900.

You are given a Roman numeral. Convert it to its integer value.

Input

The first line contains an integer n $(1 \le n \le 21)$ — the length of the Roman numeral.

The second line contains a string s of length n, consisting of uppercase letters:

$$I, V, X, L, C, D, M$$

It is guaranteed that s is a valid Roman numeral representing a number in the range $[1, 9999]$.

Output

Print the integer value of the Roman numeral.

Examples

standard input	standard output
3 III	3
5 LVIII	58
4 MMIV	2004

Problem H. Playing with squared

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

Ali loves mysteries and mind games! He designed a simple game for his friends.

There are x doors, numbered from 1 to x. You may choose exactly one door.

If you choose door number i, you will be sent to a room with number i^2.

You can only **stay in the mystery** if you go to a room with number strictly less than x.

Formally, how many doors i satisfy:

$$i^2 < x$$

Your task is to calculate this number for given x.

Input

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

Each of the next t lines contains an integer x ($1 \le x \le 10^{18}$) — the number of doors.

Output

For each test case, print a single integer — the number of doors that will keep you in the mystery.

Example

standard input	standard output
4 39 568 918 1000000000	6 23 30 31622

Problem I. Rainy Tree

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Baby blue`

Aleppo Garden is a beautiful place with a tree-like structure. The garden contains n trees connected as a rooted tree, where tree 1 is the root. Each tree has two branches. Initially, the lengths of both branches are 1.

When it rains on a tree, its branches grow. If the current branch lengths are x and y, then after receiving 1 liter of rainwater, the new lengths become:

$$x \to x + y, \quad y \to x$$

If a tree receives x liters of rainwater, this transformation happens x times.

Rain can fall on a path in the tree. Omar will tell you when rain falls along the path between two trees u and v. Every tree along this path receives x liters of rain.

Later, Omar may ask a question (wonder query) about the path from u to v. For each tree on this path, consider the length of its first branch. You need to find the greatest possible cut length g such that:

- g divides all first branch lengths on this path.
- g is as large as possible.

For example:

- If the first branch lengths are $[4, 8, 12]$, then $g = 4$.
- If the first branch lengths are $[4, 9, 12]$, then $g = 1$ (because 9 cannot be divided by anything larger).

You must compute the result modulo $10^9 + 7$.

Input

The first line contains two integers n and q ($2 \le n \le 10^5, 1 \le q \le 2 \times 10^5$) — the number of trees and the number of queries.

The second line contains $n - 1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$) — where p_i is the parent of tree i.

Each of the next q lines describes a query, which can be of two types:

- `1 u v x` — Rain query. Rain falls on the path from u to v, and every tree on this path receives x liters of rain ($1 \le x \le 10^4$).
- `2 u v` — Wonder query. You must find the greatest cut length on the path from u to v and output it modulo $10^9 + 7$.

Output

For each wonder query, print a single integer — the greatest cut length modulo $10^9 + 7$.

Example

standard input	standard output
5 5 1 1 2 2 2 5 1 1 4 3 6 2 1 1 1 5 5 2 2 5 1	1 21 3

Problem J. Coloring Land

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Silver`

Timon and Pumbaa were traveling together when they reached a land with n blocks. They were happy because there were many bugs to eat, but the land looked very dull. They decided to return home, bring some colors, and paint the land to make it more cheerful.

When they returned, they had m different colors. Each color can paint exactly a_i blocks per unit, and each color must be used in multiples of a_i — no more and no less per unit.

Timon and Pumbaa can only use **up to two distinct colors** to paint the entire land, but they can use each color any number of times (including zero). They want to find how many different combinations of one or two colors they can use to paint exactly n blocks.

For example, if $n = 12$ and $m = 3$ with colors $[2, 5, 7]$, here are valid combinations:

- Use only color 2: $2 \times 6 = 12$.
- Use color 2 and color 5: $2 \times 1 + 5 \times 2 = 12$.
- Use color 5 and color 7: $5 \times 1 + 7 \times 1 = 12$.

Your task is to help Timon and Pumbaa find the number of valid combinations to paint all n blocks.

Input

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

Each test case consists of:

- A line with two integers n and m $(1 \le n \le 10^6, 1 \le m \le 1000)$ — the number of blocks and the number of colors.
- A line with m integers $a_1, a_2, \ldots, a_m$ $(1 \le a_i \le 10^6)$ — the units each color can paint.

It is guaranteed that the sum of m over all test cases does not exceed 1000.

Output

For each test case, print a single integer — the number of valid ways to paint exactly n blocks using up to two colors.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>1
12 3
2 5 7</pre></td><td><pre>3</pre></td></tr>
</table>

Note

Each combination can use any number of units of each color, but no more than two distinct colors can be used.

Problem K. FCAI Expansion

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark blue`

After years, FCAI has decided to expand its campus by purchasing nearby buildings to increase its capacity. To achieve this, they want to accommodate at least x additional students. However, they have a limited budget of b.

There are n available options. Each option has a cost of c_i and provides a capacity of p_i.

Your task is to determine whether it is possible to increase the total capacity by at least x students within the budget. If possible, you should also find the best way to choose options according to the following rules:

- Among all ways to achieve at least x extra capacity, maximize the total capacity.
- If there are multiple ways with the same total capacity, choose the one with the minimal total cost.
- If there are still multiple valid ways, choose the one with the smallest number of options.

If there are multiple valid answers after applying all rules, you may print any of them.

Input

The first line contains three integers n, b, and x ($1 \le n \le 10^3, 1 \le b, x \le 10^4$) — the number of available options, the budget, and the required additional capacity.

The next n lines each contain two integers c_i and p_i ($1 \le c_i, p_i \le 10^5$) — the cost and capacity of the i-th option.

Output

If it is impossible to achieve at least x extra capacity within the budget, print:

`Keep Searching!`

Otherwise, print two lines:

- The first line should contain three integers — the total capacity achieved, the total cost, and the number of chosen options.
- The second line should contain the indices of the chosen options in any order. Options are numbered from 1 to n.

Examples

standard input	standard output
3 1000 10000 500 500 300 550 1000 999	Keep Searching!
5 8 10000 1 10000 3 500 2 300 5 700 4 500	11000 8 3 1 3 4

Problem L. The Generous Friend

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Burgundy`

Amr wants to invite his friends for dinner. He decided that he will invite them only if he has enough coins to give each friend exactly k coins.

Given n friends and Amr's total amount of coins c, determine whether he can invite them or not.

If he can, print `YES`. Otherwise, print `NO`.

Input

The first line contains two integers n and k ($1 \le n, k \le 1000$) — the number of friends and the number of coins each friend should receive.

The second line contains a single integer c ($0 \le c \le 10^6$) — the total number of coins Amr has.

Output

Print `YES` if Amr can invite all his friends, otherwise print `NO`.

Examples

standard input	standard output
3 5 20	YES
4 6 20	NO

Note

In the first example:

- $3 \times 5 = 15$ which is less than or equal to 20, so Amr can invite them.

In the second example:

- $4 \times 6 = 24$ which is greater than 20, so he cannot invite them.

Problem M. Antymonty

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Chocolate brown`

In the beautiful land of Palestine, the olive groves and hillside terraces around Nablus have flourished for centuries, standing as a symbol of harmony between nature and tradition.

Antymonty, the queen of all ants, has ventured into one such field consisting of n blocks, each with a certain height a_i. Her young ants need a perfectly flat training ground, as they cannot walk comfortably on uneven surfaces.

To help her young ants, Antymonty wants to find the widest contiguous flat area possible. She can increase or decrease the height of any block by 1 unit in each operation. She can perform at most k such operations in total.

Your task is to determine the maximum possible width of a contiguous segment (subarray) where all blocks have equal height, after performing no more than k operations.

Input

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

Each test case consists of:

- A line with two integers n and k ($1 \le n \le 10^4, 0 \le k \le 10^9$) — the number of blocks and the maximum number of operations allowed.
- A line with n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the heights of the blocks.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case, print a single integer — the maximum width of the largest contiguous flat area that can be formed after performing at most k operations.

Example

standard input	standard output
1 5 2 1 3 2 4 5	3

Note

The initial heights are: `1 3 2 4 5`.

By performing two operations:

- Increase a_1 by 1 and decrease a_2 by 1, resulting in: `2 2 2 4 5`. The flat area `2 2 2` has width 3.
- Alternatively, increase a_3 by 1 and decrease a_4 by 1, resulting in: `1 3 3 3 5`. The flat area `3 3 3` also has width 3.

In both cases, the largest possible flat area has width 3.