DIU Take-OFF Programming Contest Fall-25 [Preliminary - A Slot]

https://toph.co/c/diu-take-off-fall-25-preliminary-a-slot

Toph

Schedule

The contest will run for **3h0m0s**.

Authors

The authors of this contest are aqshanto, juhair300, Khushnorrr, rafiwho, sajib1619, shohan241, sifat91, sourov.cse, and tasnim07.

Rules

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, and C23 GCC 13.2 in this contest.

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

Notes

There are 7 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.

A. President Amiir's Eternal Target

Following the national team's dramatic departure from the World Cup, the president of the **Backbenchers Cricket Board (BCB)** appeared at a press conference armed with confidence, charisma, and an excuse he has perfected over the years.

The President, Mr. Amiir, delivered his statement proudly:

"Our target is the next World Cup!"

Journalists stared. Cameras zoomed in. Social media froze.
Some reporters even checked their calendars, just to make sure he wasn't already skipping ahead by a decade. For the BCB, this was nothing new. They have a world-class tradition of *forward-thinking*, where every disappointment today is magically rebranded as "preparation for tomorrow," and every loss is described as "valuable experience for the future." Rumor has it that the BCB already has targets for the next five World Cups — none of which include the current one.

This revolutionary strategy ensures one thing:
the BCB will **never** miss its target… because the target keeps moving forward faster than their run rate.

You are now acting as a news reporter, and your task is to **publish the president's speech as a news headline**. The headline must include only the exact statement delivered by the president, and it should be printed without any quotation marks.

And one more important thing:
Don't forget to print a **newline** '\n' at the end. BCB may skip the current World Cup, but the judge won't skip checking your output format.

Input

There will be no input for this problem.

Output

Output the line mentioned in the problem statement without the quotes.

Example

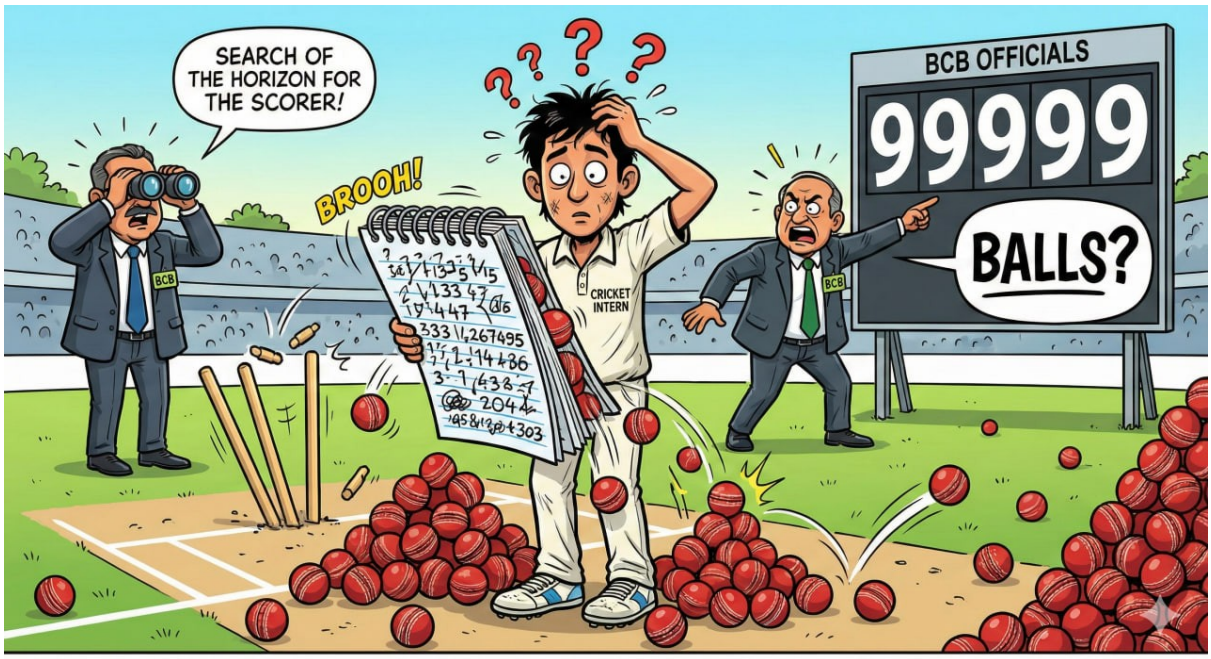
Input	Output
	`Our target is the next World Cup!`

B. BCB New Strategy

After years of blaming coaches, weather forecasts, horoscope mismatches, unlucky coin tosses, and even the stadium snacks for every defeat, the Backbenchers Cricket Board (BCB) finally decided to "fix cricket" in their own legendary style.

This time, the board's newly appointed chief strategist, Nezu, stepped forward with a revolutionary idea that he claims will shock the cricketing world:

Every over will now contain 7 balls instead of 6.



Why 7?

Because, according to Nezu, "even numbers are overrated," and also because "the opposition won't know what hit them."

No one else understood the logic, but the board approved it instantly. Because when has logic ever stopped them?

The very first match under this new rule turned into instant chaos. The scoring team got so confused that they gave up completely. Instead of recording overs and balls like normal humans, they wrote down only one thing on the score sheet, the total number of balls bowled so far. Nothing else.

Now the entire board is in panic mode. Journalists are asking questions, fans are making memes, and Nezu is confidently pretending everything is going exactly as planned.

To make sense of the mess, the board has summoned their newest intern, you, to decode the total balls into something understandable. They want the result in their brand-new format: X **overs** Y **balls**, even if it hilariously becomes something like "1 overs and 1 balls" or "0 overs and 6 balls."

Because if there is one thing the BCB truly doesn't care about…**it's grammar.**

Input

Given a single integer N $(1 \leq N \leq 10^5)$ where N is the numbers of balls.

Output

Print the result in the format "X overs Y balls" where X is the number of complete overs, and Y is the number of remaining balls after accounting for the complete overs.

Examples

Input	Output
1	0 overs 1 balls

Input	Output
11	1 overs 4 balls

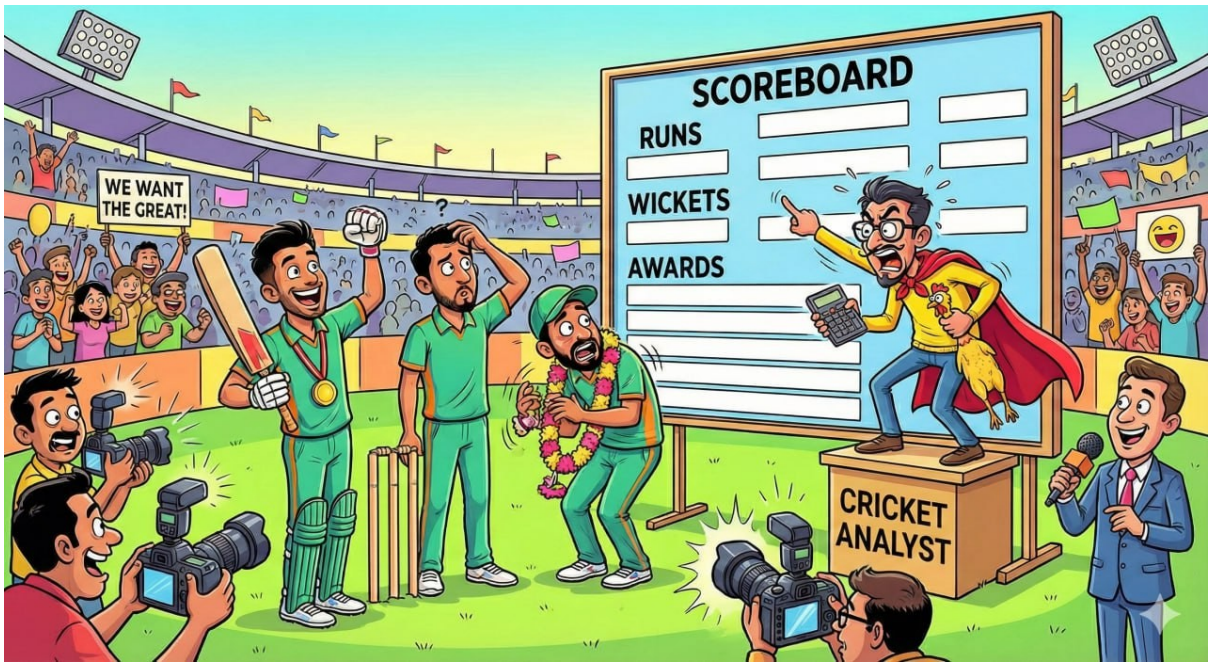
Don't forget to print a newline character '\n' at the end of your output. While the BCB may ignore grammar, but the judge will strictly check your output format.

C. Backbenchers Awards

A thrilling domestic T20 match has just ended, and the Backbenchers Cricket Board (BCB) is preparing to announce the post-match awards. Every player is nervously waiting, the right award could suddenly make them a national hero… or at least a meme.

In true BCB style, the award system is as chaotic as their strategies. Their newly appointed analyst, Shantu the Great, has devised a "mathematical but slightly confusing" rule-based system that will determine exactly which award each player gets based on just two things:

- R is the total number of runs scored by the player
- W is the total number of wickets taken by the player



Each player will receive exactly **one** award, determined solely by their **runs scored** (R) and **wickets taken** (W) in the match.

The awards are assigned as follows:

- The player receives **"Player of the Match"** if total run scored is at least 80 and also total wickets scored is at least 4.
- Otherwise, the player receives **"Best Batsman"** if total run scored is at least 80 but total wickets scored is less than 4.

- Otherwise, the player receives **"Best Bowler"** if total wickets scored is at least 4 but total run scored is less than 80.
- The player receives a **"Participation Award"** if none of the above conditions are met.

Yes, the BCB likes it complicated, but don't worry.. just follow Shantu the Great's rules exactly.

Input

The input consists of a single line containing two integers R and W

$0 \leq R \leq 200$

$0 \leq W \leq 10$

Output

Print a single line containing the player's award.

Examples

Input	Output
61 5	Best Bowler

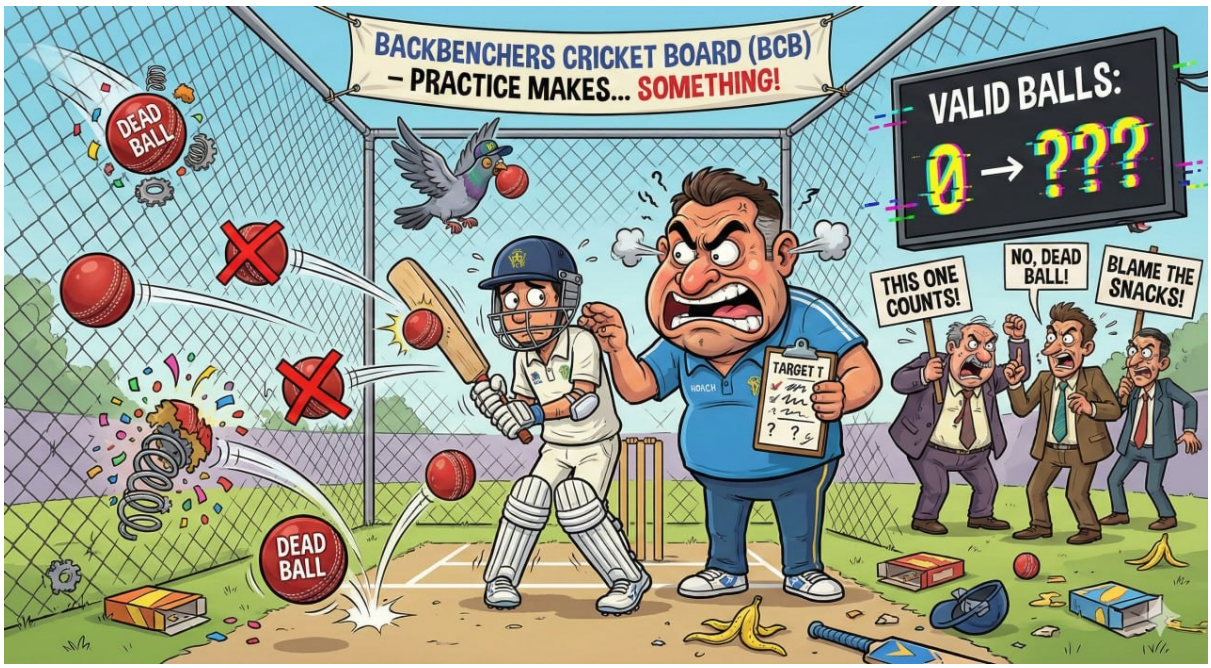
Input	Output
132 6	Player of the Match

Be careful about the newline('\n') at the end.

D. Juha's challenge

The Backbenchers Cricket Team is currently in one of its legendary “what-just-happened?” phases. To fix the batting performance once and for all, the Backbenchers Cricket Board (BCB) has tasked the head coach, Mr. Juha, with designing a “strict but slightly terrifying” training session.

In this session, Mr. Juha gives each batsman a target score of T runs. The batsman is then allowed to face N balls in the nets, where some balls might be invalid - like dead balls, net malfunctions, or mysterious BCB-approved interruptions. Invalid balls are recorded as -1 in the run log. An invalid ball will not contribute to the runs.



Mr. Juha is impatient so, he wants to know how quickly a batsman can reach the target, but only valid balls count toward reaching the score. After all, if it takes too many valid balls, the BCB can blame the batsman, the net, or even the stadium snacks.

You are given N, the total number of balls faced (including valid and invalid), and a sequence of N integers, where each number represents the runs scored on that ball. -1 indicates an **invalid** (dead) ball. Calculate the minimum valid balls needed to reach the target score **T**.

Print the number of valid balls the batsman needed to reach the target T. Once the batsman reaches T, they stop facing any more balls. If the batsman fails to reach the target print “**Mission Failed**“.

Input

- The first line contains two integers N and T ($1 \leq N \leq 10^5$, $1 \leq T \leq 10^9$) — the number of balls and the target run.
- The second line contains N integers $A_1, A_2, \ldots, A_N$ ($-1 \leq A_i \leq 10^9$) — the runs scored on each ball.

Output

- Print a single integer: the number of valid balls it took to reach or cross the target.
- If the total run after N balls is less than T, print "**Mission Failed**".

Examples

Input	Output
5 10 1 4 2 3 6	4

Input	Output
3 20 6 -1 16	2

E. BPPL - Backbenchers Premier Programmers League!

The **Backbenchers Cricket Team(BCT)** never fails to deliver premium-quality confusion. In one match, a player might hit a boundary on the first ball, and the next moment the captain shouts, *"Tension nai, everything is under control!"* and the bowler immediately delivers a wide. Classic Backbenchers behavior.

Now, welcome to the **Backbenchers Premier Programmers League (BPPL)** , a fictional tournament where the legendary Backbenchers stars are playing, and **you** are sitting in the official scorer's box with a laptop, trying to make sense of the madness unfolding ball by ball.

You are given N **overs** of ball-by-ball results. **Each over consists of 6 balls**.
Each ball can be one of the following:

- **1, 2, 3, 4 or 6** - a normal scoring shot, casually played like a Backbencher in "beast mode".
- **–1** - the batsman is **OUT!** This is the Backbenchers equivalent of a random disaster: maybe the bat slipped, maybe the ball hit a flying mosquito, maybe it was just destiny.

Whenever a **–1** appears, the current batsman is out and the next player comes in. Some might face 10 balls before getting out, some might survive only 1 ball, and some legendary Backbenchers might stay till the end, anchoring like silent warriors.

For **each batsman**, you must calculate their **run rate**, defined as:

Run Rate = Total Runs Scored / Total Balls Faced

The run rate must be printed as a **double value up to 2 decimal places**, because BCB analysts insist that "decimals build character." At the end of the innings, even if the current batsman is not out, he must still be counted.

Finally, you are given a number **K**. Print the run rate of the **first K batsmen**. If **K** is greater than the number of players who batted, print all available players and then print: "**No more players"** (Without Quotation).

Input

The first line contains two integers N and K ($1 \leq N, K \leq 10^5$) where:

- N is the number of overs in the match.
- K is the number of batsmen whose run rates need to be printed.

The next line contains $N \times 6$ space-separated integers representing the outcome of each ball (runs or wicket) : **1, 2, 3, 4, 6 or -1**

Output

Print the run rate of the first K batsmen. If the total number of batsmen is less than K, print the run rate of all available batsmen and then print: "**No more players**" (Without Quotation)

Examples

Input	Output
4 2 1 3 4 3 -1 4 2 2 1 3 6 4 4 -1 1 3 4 1 6 3 3 4 1 -1	2.20 2.89

Input	Output
7 12 1 2 4 6 -1 2 3 2 6 3 2 2 4 6 1 3 2 6 4 4 1 2 6 2 3 3 6 4 3 2 4 -1 6 1 -1 1 4 4 6 3 -1 -1	2.60 3.19 2.33 3.00 0.00 No more players

F. Backbenchers: The Last Spell

Backbenchers have reached the Cricket World Cup final. The whole nation is holding its breath. **Mr. Biswas** walks in to bat, and the stadium roars. Suddenly, he remembers something strange. Last night, in the middle of his sleep, he heard a voice reciting a spell: **"Abracadabra…"** He didn't understand it then. But now, on the biggest night of his life, his heart tells him to try it. So Mr. Biswas closes his eyes for a moment on the crease, takes a deep breath, and mutters: **"Abracadabra."**

Instantly, the spell awakens. It gives him a gift… but also a challenge.

He can choose any number K such that $1 \leq K \leq N$. If he chooses K, he will score K **runs on each ball for the next** K **balls**, then $(K-1)$ **runs on each ball for the next** $(K-1)$ **balls**, then $(K-2)$ **runs on each ball for the next** $(K-2)$ **balls**, and so on, **until** K **becomes** 0. After each ball, the runs scored are **added to the total runs**. The match ends in **two cases**: either when his total runs become greater or equal to R, or when K **becomes** 0.

From the pavilion, you must secretly send him a message. If it is possible to score exactly R **runs** using the different spell for some K $(1 \leq K \leq N)$, minimize the number of balls he faced and print R **divided by the total number of balls faced** in simplest form. That is, if R **is divisible by the total balls faced**, print the integer result. Otherwise, print the fraction in its **simplest form** by dividing both numerator and denominator by the largest number that divides them evenly. For example, if $R = 30$ and the batsman faced 12 **balls**, the output should be $5/2$.

If it is impossible to score exactly R **runs**, print -1.

Input

Given two integers $R(1 \leq R \leq 10^9)$ and $N(1 \leq N \leq 10^3)$. , Where R is the target run

Output

If it is possible to score exactly R runs, print the value of **R / total balls faced**.

If R is divisible by the total number of balls faced, print the **integer result**. Otherwise, print the fraction in the form p/q. For example, if $R = 30$ and the batsman faced 12 balls, the output should be $5/2$.

If it is impossible to score exactly R runs, print -1.

Examples

Input	Output
30 4	3

Input	Output
14 3	7/3

Input	Output
12 3	-1

G. Shift & Query

Sabbir, a dedicated student of Changaon International University, is also an unwavering supporter of the **Backbenchers cricket team**. On the evening of a much-anticipated, match at Mirpur, he proudly wore his team jersey, picked up his flag, and confidently announced to his friends: **“Ajke stadium jabo. Kew thamate parbe na.”**
However, circumstances had other intentions. The same night happened to be the final submission deadline for his assignment in the course **“Programming and Problem Solving.”** His instructor, well-known for maintaining deadlines sharper than a Taskin Ahmed Yorker, made it very clear that: **“No deadline will be extended under any circumstances, and late submissions will not be accepted.”**
Sabbir quickly realized he was trapped between academic duty and cricket passion. And since his programming skills are… *let’s just say, still developing*, there is no realistic way he can complete the assignment on his own. Therefore, he is relying entirely on your assistance to solve the following task before the deadline expires and he can finally make it to the stadium and cheer for the Tigers without academic consequences.

Task :
You are given a string S of length n, consisting of lowercase English letters **'**a**' – '**z**'** and the dot character '.' , representing an **empty space**.

You must process q operations of the following two types:
$Query\ Type$ **-** 1 **:** $1\ c$
When this query is applied to character ‘c’ :

- Every occurrence of ‘c’ attempts to jump forward **(toward the left)** into an empty position '.' .
- Character ‘c’ at ith **-** index may only jump into leftmost empty space '.' between indices 1 and i .
- Characters are processed from left to right.
- Each character moves to the leftmost available empty space on its left.
- After a character jumps, its old position becomes '.' immediately and is considered empty for subsequent movements during the same operation.

$Query\ Type$ - 2 : $2\ l\ r\ c$

- For the substring from index l to r (inclusive), count and print how many total characters are **NOT equal** to **c**.
- **Empty space ('.') does not count as character.**

Input

First line contains 2 integers n and q. n is the size of the string and q is number of total query.
Next line contains a string containing lowercase English letters and dots '.' of size n.
Next q lines contain the queries and each query in one of the two formats :

- $1\ c$
- $2\ l\ r\ c$

Constraints :
$1 \leq n \leq 2 \times 10^5$
$1 \leq q \leq 2 \times 10^5$
$1 \leq l \leq r \leq n$
'c' **is a lowercase letter ('**a**' – '**z**')**

Output

For each query of type 2 You have to print the answer in new lines (the number of characters between l to r that is not equal to character 'c').

Examples

Input	Output
11 3 bc.df.a..oa 1 a 2 1 6 c 2 5 9 z	5 2

The string has length **11**, and there are **3 queries**.
Query 1 → `1 a`

Input	Output
This is a **Type-1 query**, so all characters `'a'` will try to jump forward into empty spaces (`.`) that lie **before their current position**. • The `'a'` at index **7** sees empty spaces at indices **3, 6, 8, 9** → But it can only jump into empty spaces **between 1 and 7** → `{3, 6}` → The leftmost free spot before 7 is **index 3** → So `'a'` moves from 7 → 3 • Next `'a'` is at index **11** → Free spaces (after first move): **6, 8, 9, 7** → It can only use free spaces **between 1 and 11** → `{6, 7, 8, 9}` → Leftmost among these is **index 6** → So `'a'` moves from 11 → 6 After applying all movements (left to right), the string becomes : “bcadfa . . . o .” **Query 2** → `2 1 6 c` Characters not equal to `'c' in the range of (1 to 6)` are : **b, a, d, f, a → 5 characters** **Query 3 → `2 5 9 z`** Count how many characters in indices **5 to 9 (fa . . .**) are **NOT equal to 'z'. `Since` 'z'` does not appear at all, every character in that range is counted. and only 2 character f , a apears in that range (dot mean empty space so does not count)	

Input	Output
21 6 ..xy...bc.coefaa.c.oa 2 1 4 y 1 o 2 12 15 c 1 x 1 a 2 12 15 c	1 3 2