



# Northeastern Programming Marathon

Problem Booklet

2026

## General Information

This problem booklet has 25 pages (not counting the cover), numbered from 1 to 25. Please check that the booklet is complete.

### Input

- Input must be read from standard input.
- Every input line, including the last one, ends with a newline character (`\n`).
- When an input line contains several space-separated values, there is exactly one whitespace character between two consecutive values on the same line.

### Output

- Output must be written to standard output.
- Output must follow the format specified in the problem statement. It must not contain any extra data.
- Every output line, including the last one, must end with a newline character (`\n`).
- When an output line displays several space-separated values, there must be exactly one whitespace character between two consecutive values.
- When an output value is a real number, use at least the number of decimal places required by the precision stated in the problem statement.

### Interactive Problems

The contest may contain interactive problems. In this type of problem, the input data given to your program may not be predetermined; instead, it is built specifically for your solution. The judge runs a special program (the interactor), whose output is forwarded to your solution's input, and your program's output is sent to the interactor's input. In other words, your solution and the interactor exchange data and may decide what to print based on the communication history.

When writing a solution for an interactive problem, remember that anything you print may first be stored in an internal *buffer* and not immediately transferred to the interactor. To avoid this, you must use a special *flush* operation whenever you print data. Flush operations are available in the standard libraries of almost every programming language:

- `fflush(stdout)` in C.
- `cout.flush()` in C++.
- `sys.stdout.flush()` in Python.
- `System.out.flush()` in Java and Kotlin.

# Problem A. Arretada Removals

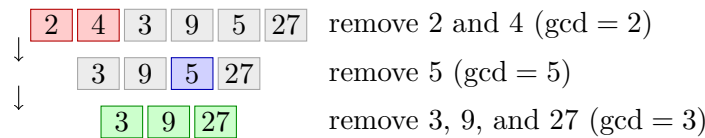
Time limit: 1000 ms  
Memory limit: 1024 MiB

The Tengo-lengo-tengo Football Federation (FFT) is organizing the Tengo-lengo-tengo World Cup, a tournament unlike anything ever seen before. For the opening show, the organizers placed a line of  $N$  flags on the field. The  $i$ -th flag represents one of the teams in the cup and has an integer  $A_i$  written on it.

During the show, flags will be removed from the field through *arretada* removals. A removal is called *arretada* when the opening crew chooses a non-empty segment of consecutive flags, say from position  $L$  to position  $R$ , such that:

$$\gcd(A_L, A_{L+1}, \dots, A_R) > 1$$

Here,  $\gcd$  denotes the greatest common divisor of the involved numbers. In an *arretada* removal, all flags in that segment are removed. After that, the remaining flags close the gap, keeping the same relative order as before.



One possible sequence of removals for the second sample. In each row, the highlighted flags are removed.

To plan the opening show properly, the FFT wants to know the minimum number of *arretada* removals needed to remove all flags during the presentation.

## Input

The first line contains an integer  $N$  ( $1 \leq N \leq 200$ ), the number of flags.

The second line contains  $N$  integers  $A_1, A_2, \dots, A_N$  ( $2 \leq A_i \leq 10^6$ ), the numbers written on the flags.

## Output

Print a single integer: the minimum number of *arretada* removals needed to remove all flags.

## Examples

| Sample input 1    | Sample output 1 |
|-------------------|-----------------|
| 6<br>2 3 5 5 3 2  | 3               |
| Sample input 2    | Sample output 2 |
| 6<br>2 4 3 9 5 27 | 3               |

## Problem B. Good Spotlights

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

In the final of the Tengo-lengo-tengo Football World Cup, the stadium light show is the highlight of halftime while several artists perform on stage. Each spotlight configuration is represented by a binary number with exactly 3 bits: bit 0 means a light is off, and bit 1 means a light is on.

To make the halftime show look smooth, the organizers want the lights to change calmly. Therefore, from one configuration to the next, exactly one spotlight must change state.

You are given an initial configuration  $X_0$ . Your task is to produce a sequence that starts at  $X_0$ , visits all 8 configurations of 3 bits exactly once, and then returns to  $X_0$ .

### Input

The input contains a single line with a string  $X_0$  of exactly 3 characters. Each character of  $X_0$  is either 0 or 1.

### Output

Print 9 binary strings of exactly 3 bits.

The first and the last string must be equal to  $X_0$ . Among the first 8 strings, each of the 8 possible configurations must appear exactly once. Also, any two consecutive strings in your output must differ in exactly one bit.

If there is more than one valid answer, print any of them.

### Examples

| Sample input 1 | Sample output 1                                             |
|----------------|-------------------------------------------------------------|
| 010            | 010<br>110<br>111<br>101<br>100<br>000<br>001<br>011<br>010 |

# Problem C. Youth Divisions

*Time limit: 2000 ms*  
*Memory limit: 1024 MiB*

A football club is reorganizing its youth divisions to take part in a championship organized by the FFT, the Tongo-lengo-tengo Football Federation.

There are  $N$  available professionals, and each professional has a main role in a coaching staff.

Roles are represented by integers. For example, role 1 may represent a head coach, role 2 a fitness coach, role 3 a physiotherapist, role 4 a performance analyst, and role 5 a goalkeeper coach.

The  $i$ -th professional has role  $A_i$ .

The club wants to divide all  $N$  professionals into exactly  $K$  coaching staffs, one for each youth division.

However, to avoid repeated responsibilities within the same staff, a coaching staff cannot contain two professionals with the same role. Also, every coaching staff must contain at least one professional.

Two divisions are considered different if there exists some group of professionals that forms a coaching staff in one division but does not form a coaching staff in the other. The order of the coaching staffs does not matter, and the order of professionals inside the same staff does not matter either.

Determine the number of different ways to form the  $K$  coaching staffs, modulo 998244353.

## Input

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

The second line contains  $N$  integers  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq N$ ), where  $A_i$  represents the main role of the  $i$ -th professional.

## Output

The output must contain a single integer, the number of different ways to divide the professionals into exactly  $K$  coaching staffs, modulo 998244353.

## Examples

| Sample input 1   | Sample output 1 |
|------------------|-----------------|
| 5 2<br>1 2 2 1 3 | 4               |

### Explanation of sample 1

There are 5 professionals and 2 coaching staffs must be formed. The roles are:

- Role 1: 2 professionals;
- Role 2: 2 professionals;
- Role 3: 1 professional.

One valid division is to put one professional with role 1 and one professional with role 2 in one

coaching staff, and the remaining professionals in the other coaching staff.

No coaching staff contains two professionals with the same role, and every professional is used exactly once.

**Sample input 2****Sample output 2**

|              |   |
|--------------|---|
| 3 3<br>1 2 3 | 1 |
|--------------|---|

**Explanation of sample 2**

There are 3 professionals, all with different roles, and 3 coaching staffs must be formed. The only valid division is to put one professional in each staff.

# Problem D. Crazy Decoder

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

The year is 3026, and the artificial intelligence GaLaDOS is about to take over the world. Fortunately, humanity has one last chance: it was discovered that the AI shared an administrator password with its allies. If this password is recovered, the AI can be shut down and humanity will be saved.

Of course, things will not be that easy. First, GaLaDOS defined two numbers,  $N$  and  $K$ . Then the password  $X$ , initially a number between 0 and  $2^{32} - 1$ , went through the following encryption process  $N$  times:

1. Let  $Y$  be the value of  $X \oplus (X \cdot 2^K)$ .
2. Change  $X$  to be the value of the first 32 bits of  $Y$ .

Here,  $\oplus$  represents the XOR operation\* between two numbers.

Your goal is: given  $N$ ,  $K$ , and the final result of the encryption  $S$ , build a decoder that returns the original password  $X$ . If there is more than one valid password, you may return any of them.

\* The XOR operation between two numbers is the result of the exclusive OR between the bits of their binary representations. For example:  $5 \oplus 3 = (101)_2 \oplus (011)_2 = (110)_2 = 6$ .

## Input

The input is a single line containing three integers  $N$  ( $1 \leq N \leq 10^6$ ),  $K$  ( $1 \leq K \leq 30$ ), and  $S$  ( $1 \leq S < 2^{32}$ ).

## Output

The output must be a single integer, the original password  $X$ , printed as an **unsigned 32-bit integer**.

## Examples

| Sample input 1 | Sample output 1 |
|----------------|-----------------|
| 2 2 51         | 3               |

### Explanation of sample 1

In this example,  $N = 2$ . If we start with  $X = 3$ , in the first step we get  $Y = 3 \oplus (3 \cdot 2^2) = 3 \oplus 12 = 15$ , and the value of  $X$  is updated.

In the second step, we get  $Y = 15 \oplus (15 \cdot 2^2) = 15 \oplus 60 = 51$ . The value of  $X$  is updated again, and the process ends with  $X = 51$ , which is the value of  $S$ .

| Sample input 2 | Sample output 2 |
|----------------|-----------------|
| 2 3 6          | 2249744774      |

### Explanation of sample 2

Remember that the password is a number between 0 and  $2^{32} - 1$ .

**Sample input 3**

|                |
|----------------|
| 1 1 4294967295 |
|----------------|

**Sample output 3**

|            |
|------------|
| 1431655765 |
|------------|

**Explanation of sample 3**

Remember that  $S$  is a number between 1 and  $2^{32} - 1$ .

# Problem E. The Scale Riddle

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

**This is an interactive problem. You can read more about this type of problem on page 1 of the complete problem booklet.**

At the Caruaru Fair, famous for its clothing stalls, a vendor set aside  $N$  shirts that look identical for a challenge. Among these shirts,  $N - 1$  have exactly the same weight, while one different shirt is heavier than all the others.

You may use a two-pan balance scale. In each weighing, you choose the same number of shirts to place on the left pan and on the right pan. The scale tells whether the left pan was heavier, the right pan was heavier, or both pans had the same weight.

Your goal is to find which shirt is the heaviest using at most 8 weighings.

## Interaction

Initially, the interactor prints a line containing an integer  $N$  ( $2 \leq N \leq 2000$ ), the number of shirts. The shirts are numbered from 1 to  $N$ .

The interactor is **non-adaptive**: the heaviest shirt is chosen before the interaction starts and does not change during execution.

Your program may perform weighings. To perform a weighing, print a line in the format

? t a\_1 a\_2 ... a\_t b\_1 b\_2 ... b\_t

where  $t \geq 1$ . The shirts  $a_1, a_2, \dots, a_t$  will be placed on the left pan, and the shirts  $b_1, b_2, \dots, b_t$  will be placed on the right pan. All these  $2t$  indices must be distinct and between 1 and  $N$ .

The interactor will answer with a line containing a single character:

- E, if the left pan was heavier;
- D, if the right pan was heavier;
- I, if both pans had the same weight.

When your program knows which shirt is the heaviest, it must print a line in the format

! p

where  $p$  is the index of the heaviest shirt.

The answer operation does not count as a weighing. Only operations that start with `?` are counted, and your program may perform at most 8 of them.

After printing any line, your program must *flush* the output. For example, in C++ you may use `cout << endl;` or `cout.flush()`.

If your program prints an invalid value, performs more than 8 weighings, or answers an incorrect shirt, it will receive the verdict *NO - Wrong answer*.

## Examples

| Read | Sample interaction 1 | Write |
|------|----------------------|-------|
| 4    |                      |       |
|      | ? 2 1 2 3 4          |       |
| D    |                      |       |
|      | ? 1 3 4              |       |
| E    |                      |       |
|      | ! 3                  |       |

### Explanation of sample 1

In this example, the heaviest shirt is number 3. The program uses two of the 8 allowed weighings to determine which shirt is the heaviest.

In the first weighing, shirts 1 and 2 are placed on the left pan, while shirts 3 and 4 are placed on the right pan. Since the right pan becomes heavier, the different shirt must be either 3 or 4.

In the second weighing, the program compares shirts 3 and 4. Since the left pan becomes heavier, the program concludes that shirt 3 is the heaviest.

If, in the second weighing, the right pan had been heavier, shirt 4 would be the heaviest.

It is not necessary to use all 8 allowed weighings. The program may answer as soon as it determines the index of the heaviest shirt.

# Problem F. Escaping the Sun

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

During his vacation in Aracaju, Samuell decided to spend the morning swimming at Atalaia Beach, one of the city's postcards, famous for its beautiful and long stretch of sand.

When it was time to leave, a problem came up: with the sun high in the sky, he must cross the entire beach, from the sea to the boardwalk, and request an Uber. To avoid getting sunburned, Samuell plans to use the shadows of the beach umbrellas spread across the sand and choose a route that minimizes the distance walked under the sun.

Formally, the beach can be represented by the strip between two parallel lines: one is the line where the sea meets the sand ( $y = 0$ ), and the other is the boardwalk ( $y = H$ ), at distance  $H$  from the first one. There are  $N$  beach umbrellas; the  $i$ -th one casts a circular shadow with center  $(x_i, y_i)$  and radius  $r_i$ .

Assume Samuell can swim freely underwater without being exposed to the sun, and can leave the water at any point on the sea line. Once on the sand, he can:

- walk under the shadow of a beach umbrella, including moving directly to another one when their shadows touch or overlap;
- leave the sand at any point on the boardwalk.

Help Samuell determine the minimum distance he must walk under the sun to reach the boardwalk.

## Input

The first line contains two integers  $N$  and  $H$  ( $0 \leq N \leq 1000$ ,  $1 \leq H \leq 10^5$ ), respectively the number of beach umbrellas and the distance between the sea line and the boardwalk.

Each of the next  $N$  lines contains three integers  $x_i$ ,  $y_i$ , and  $r_i$  ( $0 \leq x_i \leq 10^5$ ,  $0 \leq y_i \leq H$ ,  $1 \leq r_i \leq 10^5$ ), the center and radius of the  $i$ -th beach umbrella. A beach umbrella's shadow may extend beyond the sea line or the boardwalk.

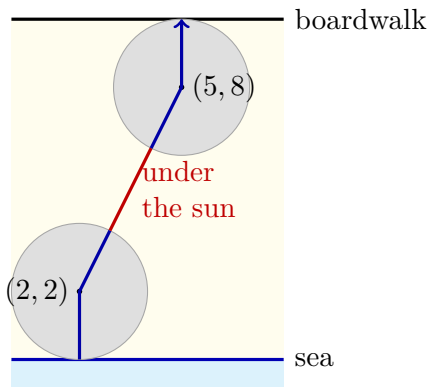
## Output

Print a single real number: the minimum distance under the sun that Samuell must walk to cross the sand, from the sea to the boardwalk.

Your answer will be considered correct if its absolute or relative error with respect to the expected answer is at most  $10^{-6}$ .

## Examples

| Sample input 1         | Sample output 1 |
|------------------------|-----------------|
| 2 10<br>2 2 2<br>5 8 2 | 2.7082039325    |

**Explanation of sample 1**

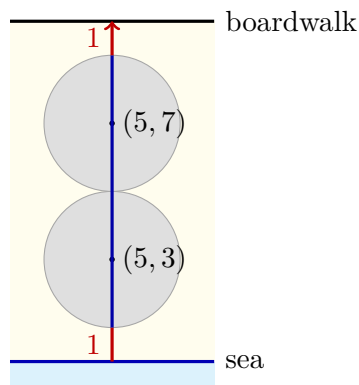
In the first example, Samuel can take a route where he walks  $3\sqrt{5} - 4$  under the sun.

**Sample input 2**

```
2 10
5 3 2
5 7 2
```

**Sample output 2**

```
2
```

**Explanation of sample 2**

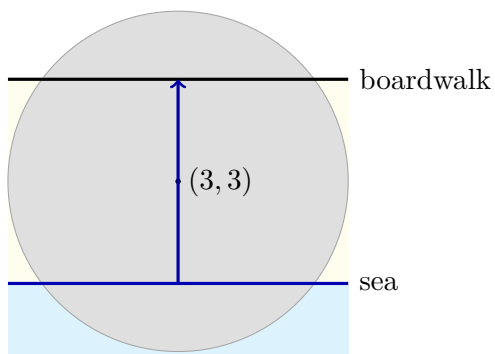
In the second example, Samuel walks one unit under the sun to reach the first shadow. The two shadows touch, allowing him to move from one to the other without being exposed. Then he walks one more unit to the boardwalk. Therefore, the total distance under the sun is  $1 + 1 = 2$ .

**Sample input 3**

```
1 6
3 3 5
```

**Sample output 3**

```
0
```

**Explanation of sample 3**

In the third example, the only shadow has radius 5 and center (3, 3). Since it reaches both the sea line and the boardwalk, Samuel can cross the whole beach while staying in the shadow. Therefore, the distance walked under the sun is 0.

# Problem G. Queue Mischief

*Time limit: 3000 ms*  
*Memory limit: 1024 MiB*

The grand final of the Tingo-lengo-tengo World Cup is about to begin! The decisive match will be played between **Altos** and **Bahia**, and the line to enter the stadium is pure chaos.

The line is made entirely of fans of the two teams. As tradition dictates, whenever an Altos fan is standing before (closer to the stadium entrance than) a Bahia fan, we say there is a *banter* happening between them.

Mathematically, if we view the line as a string  $S$  made of the letters 'A' and 'B', a *banter* is formed by each pair of indices  $(i, j)$  such that  $i < j$ , where  $S[i]$  is an Altos fan ('A') and  $S[j]$  is a Bahia fan ('B').

The stadium organization recorded  $Q$  events in the line, which can be of 4 types:

- **Type 1:** A fan of team  $X$  ('A' or 'B') joins the end of the line.
- **Type 2:** A mischievous fan of team  $X$  ('A' or 'B') cuts in at the exact front of the line.
- **Type 3:** The fan at the end of the line gets tired of waiting and leaves.
- **Type 4:** The fan at the front of the line goes through the turnstile and enters the stadium.

Your task is to compute the total number of *banters* happening in the line immediately after each event.

## Input

The first line of the input contains an integer  $Q$  ( $1 \leq Q \leq 10^6$ ), representing the number of events.

The next  $Q$  lines describe the events. Each line starts with an integer  $T$  ( $1 \leq T \leq 4$ ):

- If  $T = 1$ , it is followed by a character  $X$  ('A' or 'B'), indicating the team of the fan who joined the end of the line.
- If  $T = 2$ , it is followed by a character  $X$  ('A' or 'B'), indicating the team of the fan who joined the front of the line.
- If  $T = 3$ , the fan at the end of the line leaves (no character is given).
- If  $T = 4$ , the fan at the front of the line enters the stadium (no character is given).

It is guaranteed that operations of type 3 and 4 are never called when the line is empty.

## Output

For each query, print a single integer on one line: the number of banters happening in the line at the current moment.

## Examples

| Sample input 1 | Sample output 1 |
|----------------|-----------------|
| 1<br>2 A       | 0               |

**Sample input 2**

```
5
1 A
1 B
1 B
1 A
1 B
```

**Sample output 2**

```
0
1
2
2
4
```

**Explanation of sample 2**

In the second example, after each event the line becomes, respectively, **A**, **AB**, **ABB**, **ABBA**, and **ABBAB**. The numbers of banter in these lines are 0, 1, 2, 2, and 4, because each banter corresponds to an **A** that appears before a **B**.

**Sample input 3**

```
5
1 A
2 B
2 A
3
4
```

**Sample output 3**

```
0
0
1
1
0
```

**Explanation of sample 3**

In the third example, after each event the line becomes, respectively, **A**, **BA**, **ABA**, **AB**, and **B**. Thus, the numbers of banter are 0, 0, 1, 1, and 0.

# Problem H. Cordel Stories

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

Since childhood, the cordel poet Severino has kept tales heard at countryside markets: singing duels, enchanted creatures, and brave deeds passed down through generations. Now he is preparing his greatest booklet, an adventure where, in each chapter, the reader chooses how the story continues. To avoid losing any ending, Severino spread the pages of his outline on the floor and connected each choice with a piece of string.

Seen from above, the outline resembles a tree structure: there are  $N$  chapters numbered from 1 to  $N$ , with chapter 1 being the beginning of every story. The outline follows these rules:

- Each connection links a chapter to one of its immediate continuations;
- For each connection, Severino must choose an uppercase letter between A and Z;
- No chapter may have two continuations with the same letter;
- Each chapter with no continuations ends a complete story, described by the sequence of letters on the path from chapter 1 to it.

Severino also noticed that changing letters in the outline does not always change the booklet: two assignments of letters are considered equivalent when they produce exactly the same set of complete stories.

Severino wants to know how many distinct sets of complete stories can be obtained from this outline among all possible assignments of letters. Since the answer can be very large, he asks you to report it modulo 998244353.

## Input

The first line contains an integer  $N$  ( $2 \leq N \leq 10^5$ ), the number of chapters.

Each of the next  $N - 1$  lines contains two integers  $U$  and  $V$  ( $1 \leq U, V \leq N$ ), indicating that there is a connection between chapters  $U$  and  $V$ .

It is guaranteed that these connections form a tree, rooted at chapter 1.

## Output

Print a single integer: the number of distinct sets of complete stories that can be obtained, modulo 998244353.

## Examples

| Sample input 1  | Sample output 1 |
|-----------------|-----------------|
| 3<br>1 2<br>1 3 | 325             |
| Sample input 2  | Sample output 2 |
| 2<br>2 1        | 26              |

# Problem I. Incredible XORlandia

*Time limit: 2000 ms*  
*Memory limit: 1024 MiB*

XORlandia is a city on the coast of Northeastern Brazil, famous for its crystal-clear beaches, its enormous waterfront, and its slogan *XORlandia is massa*. Legend says that, during the construction of the city, an old fisherman known as Old Sururu cursed some of the region's streets after losing a dominoes match to a local resident.

Since then, each street in the city has had an energy, represented by an integer associated with it. The city has  $N$  tourist spots numbered from 1 to  $N$ ; spot 1 is Marco dos XORais and spot  $N$  is where the XORgante Chair is located. They are connected by  $M$  bidirectional streets. The  $i$ -th street connects spots  $U_i$  and  $V_i$  and has energy  $W_i$ . It is also known that it is possible to reach any spot from any other, and that there is at most one street connecting each pair of spots.

As a local tradition, residents define the value of a path as the XOR of the energies of all streets traversed. A path may pass through the same spot and traverse the same street multiple times. Each time a street is traversed, its energy is included again in the XOR calculation, even if that same street has already been used earlier in the path. The city hall considers XORlandia an incredible city if every path that starts at Marco dos XORais and ends at the XORgante Chair has value equal to 0.

Therefore, the city hall decided that it may choose **exactly one street** and change its energy to any other non-negative integer value, as long as it is different from the previous value.

For each street, determine whether it is possible to change only the value of that street so that the city becomes incredible.

## Input

The first line contains two integers  $N$  and  $M$ , indicating respectively the number of tourist spots in XORlandia and the number of streets.

Each of the next  $M$  lines contains three integers  $U_i$ ,  $V_i$ , and  $W_i$  ( $1 \leq U_i, V_i \leq N$ ,  $U_i \neq V_i$ ,  $0 \leq W_i < 2^{30}$ ), representing a bidirectional street connecting spots  $U_i$  and  $V_i$ , with energy  $W_i$ .

- $2 \leq N \leq 200000$
- $N - 1 \leq M \leq \min\left(\frac{N(N-1)}{2}, 200000\right)$

## Output

Print a binary string of length  $M$ .

The  $i$ -th character must be:

- 1, if it is possible to modify only the  $i$ -th street so that XORlandia becomes incredible;
- 0, otherwise.

## Examples

| Sample input 1                                   | Sample output 1 |
|--------------------------------------------------|-----------------|
| 5 5<br>1 2 1<br>1 3 0<br>3 4 0<br>2 4 2<br>4 5 3 | 01100           |

### Explanation of sample 1

In the first example, it is possible to change the energy of the second or third street so that every path from spot 1 to spot 5 has value 0. None of the other streets allows this condition to be achieved with a single change. Therefore, the answer is 01100.

| Sample input 2            | Sample output 2 |
|---------------------------|-----------------|
| 3 2<br>1 2 127<br>2 3 127 | 00              |

### Explanation of sample 2

In the second example, the path from spot 1 to spot 3 has value  $127 \oplus 127 = 0$ . However, since it is mandatory to change the energy of exactly one street to a different value, it is impossible to keep this value equal to 0. Therefore, the answer is 00.

| Sample input 3            | Sample output 3 |
|---------------------------|-----------------|
| 3 2<br>1 2 127<br>2 3 128 | 11              |

### Explanation of sample 3

In the third example, one may change the energy of the first street to 128 or the energy of the second street to 127. In either choice, the path from spot 1 to spot 3 has value 0. Therefore, the answer is 11.

# Problem J. The Brega Game

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*



Reginaldo and Adilson will perform on several days during a brega music festival in the countryside of Pernambuco. To decide which songs each of them will sing, they created a game, affectionately nicknamed *The Brega Game*.

In this game, there is a sequence of  $N$  cards placed at the center of a table. Each card has a distinct number from 1 to  $N$ , so the sequence forms a permutation of these values. Each card represents a song in the repertoire, and the number written on it is that song's score, indicating how much the audience likes it.

The festival will have several performance days, and on each day they choose an interval of the sequence, indicating which songs will be available to sing that day. The game is then played considering only this contiguous interval of cards. The singers alternate turns, with Adilson always making the first move. On each turn, the current player must remove exactly one card from the remaining sequence: either the leftmost card or the rightmost card. The removed card represents a song that will be sung by that singer during the show.

Each singer's score is the largest number written among all cards they removed during the game, representing the song by that singer that excited the audience the most. If a singer does not remove any card, their score is 0. When all cards have been removed, the singer with the larger score wins. Since all numbers in the sequence are distinct, there is always exactly one winner.

For each given interval, determine which singer wins assuming both play optimally.

## Input

The first line contains two integers  $N$  and  $Q$  ( $1 \leq N \leq 2 \times 10^5$ ,  $1 \leq Q \leq 2 \times 10^5$ ), respectively the number of songs in the repertoire and the number of festival days.

The second line contains  $N$  integers  $P_1, P_2, \dots, P_N$ , a permutation of the integers from 1 to  $N$ .

For each  $1 \leq i \leq Q$ , the  $i$ -th of the next  $Q$  lines contains two integers  $L_i$  and  $R_i$  ( $1 \leq L_i \leq R_i \leq N$ ), indicating that on that festival day only the songs  $P_{L_i}, P_{L_i+1}, \dots, P_{R_i}$  will be sung.

## Output

For each festival day, print a line containing the name of the winning singer: **Reginaldo**, if Reginaldo wins, or **Adilson**, if Adilson wins.

## Examples

| Sample input 1                        | Sample output 1                 |
|---------------------------------------|---------------------------------|
| 5 3<br>2 5 1 4 3<br>1 5<br>1 1<br>2 4 | Reginaldo<br>Adilson<br>Adilson |

### Explanation of sample 1

On the first day, all cards are used, and Reginaldo wins when both play optimally. On the second day, there is only card 2, which Adilson removes on the first move. On the third day, the cards are 5, 1, and 4, and Adilson can remove card 5 immediately.

# Problem K. Karamelos at São João

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

João went to the São João festival in Caruaru with a simple mission, at least in his mind: buy sweets for his friends and still keep a little treat for himself to eat after the forró.

João has  $A$  friends and found a stall selling exactly  $X$  different types of sweets. He wants to choose a positive integer  $Y$  and buy exactly  $Y$  sweets of each type. Thus, in total, João would buy  $X \cdot Y$  sweets.

After the purchase, João will distribute the sweets equally among his  $A$  friends, possibly leaving some sweets over. Each friend must receive at least one sweet, because showing up at the party and leaving empty-handed is reason for jokes until the next São João.

To make the count come out exactly as he imagined, exactly one sweet must be left for João himself. Therefore, he wants to know whether there exists some value of  $Y$  that allows this. If it exists, determine the smallest possible one.

## Input

The input contains two integers  $A$  and  $X$  ( $2 \leq A \leq 10^4$  and  $1 \leq X < A$ ), the number of friends and the number of types of sweets.

## Output

Print the smallest possible value of  $Y$ , or  $-1$  if it does not exist.

## Examples

| Sample input 1 | Sample output 1 |
|----------------|-----------------|
| 5 2            | 3               |

### Explanation of sample 1

With  $Y = 3$ , João buys 6 sweets: he gives 1 sweet to each of his 5 friends and keeps the remaining sweet.

| Sample input 2 | Sample output 2 |
|----------------|-----------------|
| 6 4            | -1              |

### Explanation of sample 2

The numbers 6 and 4 have 2 as their greatest common divisor. Therefore, every total purchase is a multiple of 4, and when divided among 6 friends, the remainder is never 1. Thus, there is no valid value of  $Y$ .

| Sample input 3 | Sample output 3 |
|----------------|-----------------|
| 7 1            | 8               |

### Explanation of sample 3

With  $Y = 1$ , there would be 1 sweet left over, but no friend would receive a sweet. The smallest valid value is  $Y = 8$ .

# Problem L. Lampions League

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

Every four years, in Northeastern Brazil, the FFT (Tengo-lengo-tengo Football Federation) holds the Tengo-lengo-tengo World Cup. Because of the great importance of this competition, the fans soon affectionately nicknamed the tournament the Lampions League, in reference to that weaker European competition; after all, everyone knows the competitive level in the Tengo-lengo-tengo World Cup is high.

During the final match, the official announcer will narrate the plays in a special way. On the first play, he shouts **Tengo** 1 time; on the second, he shouts **Lengo** 2 times; on the third, he shouts **Tengo** 3 times; on the fourth, he shouts **Lengo** 4 times; and so on, alternating the word until play  $N$ .

The FFT needs to know whether the absolute difference between the number of times **Tengo** was shouted and the number of times **Lengo** was shouted is even or odd.

Given the value of  $N$ , determine whether this difference is even or odd.

## Input

The input contains a single line with an integer  $N$  ( $1 \leq N \leq 10^4$ ), representing the number of narrated plays.

## Output

The output must contain a single line with the word **PAR**, if the absolute difference between the number of shouted **Tengos** and **Lengos** is even, or **IMPAR** if this difference is odd.

## Examples

| Sample input 1 | Sample output 1 |
|----------------|-----------------|
| 2              | IMPAR           |
| Sample input 2 | Sample output 2 |
| 10000          | PAR             |

# Problem M. The World's Biggest São João

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

Every year, one debate stirs Northeastern Brazil: after all, which is the world's biggest São João, the one in Campina Grande or the one in Caruaru?

To try to settle the matter, a research institute decided to run two polls: one during the São João festival in Campina Grande and another during the São João festival in Caruaru.

The  $N$  interviewed people attended both festivals, experiencing both Campina Grande's São João and Caruaru's São João up close. However, for organizational reasons, each person took part in exactly one of the two polls.

It is allowed for one of the polls to be empty.

Each person has a true opinion about which festival they prefer: Campina Grande or Caruaru.

However, the polls happened in the middle of the party, with lots of forró, fireworks, and noise. Because of that, when a person said their answer, the institute could end up hearing it wrong. Thus, the vote recorded for a person may be different from their true opinion.

A few days later, each participant receives the result of the poll they joined: how many votes were recorded for Campina Grande and how many votes were recorded for Caruaru.

Each person's evaluation of the poll depends only on their true opinion:

- for each vote recorded for the festival they prefer, their evaluation increases by 1;
- for each vote recorded for the other festival, their evaluation decreases by 1.

The total score of a poll is the sum of the evaluations of all its participants.

Assuming each participant was assigned to a poll in a way that maximizes the sum of the scores, determine the largest possible sum of the scores of the two polls.

## Input

The first line contains an integer  $N$  ( $1 \leq N \leq 2 \times 10^5$ ), the number of interviewed people.

The second line contains a string  $P$  of length  $N$ , where  $P_i$  represents the true opinion of person  $i$ :

- G indicates that they prefer Campina Grande's São João;
- R indicates that they prefer Caruaru's São João.

The third line contains a string  $V$  of length  $N$ , where  $V_i$  represents the vote that was recorded for person  $i$ :

- G indicates that the vote was recorded for Campina Grande;
- R indicates that the vote was recorded for Caruaru.

## Output

The output must contain only one integer, the largest possible sum of the scores of the two polls.

## Examples

| Sample input 1    | Sample output 1 |
|-------------------|-----------------|
| 4<br>GRRR<br>GRGR | 2               |

### Explanation of sample 1

One optimal division is to put person 1 in the first poll and people 2, 3, and 4 in the second poll.

In the first poll, there is only person 1. Their recorded vote was G, and they prefer G, so their evaluation is 1. Therefore, the score of the first poll is 1.

In the second poll, the recorded votes of people 2, 3, and 4 are, respectively, R, G, and R. Thus, the result of this poll contains 1 vote for Campina Grande and 2 votes for Caruaru.

Person 2 prefers G, so their evaluation is  $1 - 2 = -1$ .

Person 3 prefers R, so their evaluation is  $2 - 1 = 1$ .

Person 4 prefers R, so their evaluation is  $2 - 1 = 1$ .

Therefore, the score of the second poll is  $-1 + 1 + 1 = 1$ . The sum of the two poll scores is  $1 + 1 = 2$ , which is the maximum possible value.

| Sample input 2  | Sample output 2 |
|-----------------|-----------------|
| 3<br>GGG<br>GGG | 9               |

### Explanation of sample 2

Everyone prefers Campina Grande, and all votes were recorded for Campina Grande.

One optimal division is to put all people in the same poll and leave the other poll empty.

In the poll with the three people, there are 3 votes recorded for Campina Grande and 0 votes recorded for Caruaru. Since everyone prefers Campina Grande, each of them has evaluation  $3 - 0 = 3$ .

Therefore, the score of this poll is  $3 + 3 + 3 = 9$ . The other poll is empty and contributes 0 points.

Thus, the largest possible sum of scores is 9.

# Problem N. Did You See My Album?

*Time limit: 1000 ms*  
*Memory limit: 1024 MiB*

Xand Avião is very excited to start completing the new sticker album of the Tengo-lengo-tengo Football Federation (FFT). That is because the new FFT album works like bingo: on the day you complete it, you win a prize!

The album initially has  $N$  ( $N \leq 2026$ ) players licensed by the FFT. Each day, in the morning, there is a  $P\%$  chance that a new player joins the album. This can only happen if there are fewer than 2026 players in the album. After all, this is the total number of football players in the world!

Xand initially has no stickers. So every day, in the afternoon, Xand will buy a new one. The sticker is equally likely to be any player currently in the album. At that moment, if Xand has all stickers of the players that are already in the album, he immediately exchanges the album for his prize.

Xand is so excited about the prize that he can barely count the days until he completes the album. So he asked for your help to count for him: what is the expected number of days Xand will need to win the sticker album prize?

## Input

The input is a single line containing two integers,  $N$  ( $1 \leq N \leq 2026$ ) and  $P$  ( $0 \leq P \leq 100$ ).

## Output

Your output must contain a single number indicating the expected number of days Xand will need to complete the sticker album. Your answer will be considered correct if it has relative or absolute error at most  $10^{-6}$  with respect to the judge’s answer.

## Examples

| Sample input 1 | Sample output 1 |
|----------------|-----------------|
| 1 0            | 1.000000000000  |

### Explanation of sample 1

On the first day, it is guaranteed that no new player will be added, and Xand will get the only sticker he needs.

| Sample input 2 | Sample output 2 |
|----------------|-----------------|
| 2 0            | 3.000000000000  |

### Explanation of sample 2

It is guaranteed that no new player will be added. Therefore, Xand has probability  $\frac{1}{2}$  of completing the album in 2 days,  $\frac{1}{4}$  of completing it in 3 days, and so on. Thus, the expected value is  $\sum_{i=1}^{\infty} (i + 1) \left(\frac{1}{2}\right)^i = 3$ .

| Sample input 3 | Sample output 3  |
|----------------|------------------|
| 5 10           | 203.072212695099 |

# Problem O. Operation Mandacaru

*Time limit: 5000 ms*  
*Memory limit: 1024 MiB*

Viação Mandacaru runs a daily route through  $M$  cities in Northeastern Brazil, numbered from 1 to  $M$  in the order in which they are visited. The bus used on this route has exactly  $C$  seats.

One day, a system failure simultaneously interrupted  $N$  ticket requests. Although no ticket was issued, the request data was recovered. Passenger  $i$ 's request is described by three integers  $L_i$ ,  $R_i$ , and  $P_i$ : the passenger wants to board in city  $L_i$ , get off in city  $R_i$ , and would pay  $P_i$  reais for the ticket.

The company does not allow passengers to choose their seats. Therefore, a request can only be accepted if the bus has an available seat during the entire trip between cities  $L_i$  and  $R_i$ . In each city, all passengers whose trip ends there get off before new passengers board. Thus, a passenger boarding in city  $j$  may occupy a seat freed in that same city.

Since all requests were received at the same time, it may be impossible to accept all of them without exceeding the bus capacity on some segment of the route. Always concerned with solving the situation as fairly as possible, especially for its own cash register, Viação Mandacaru decided to choose which tickets will be issued so as to maximize its revenue.

Determine the maximum total value the company can earn by issuing a subset of the requests, without the bus carrying more than  $C$  passengers simultaneously on any segment of the trip.

## Input

The first line contains three integers  $N$ ,  $M$ , and  $C$  ( $1 \leq N \leq 10^5$ ,  $2 \leq M \leq 10^5$ , and  $1 \leq C \leq 50$ ), representing, respectively, the number of requests, the number of cities on the route, and the number of seats on the bus.

Each of the next  $N$  lines contains three integers  $L_i$ ,  $R_i$ , and  $P_i$  ( $1 \leq L_i < R_i \leq M$  and  $1 \leq P_i \leq 10^9$ ), indicating that the passenger from request  $i$  wants to board in city  $L_i$ , get off in city  $R_i$ , and would pay  $P_i$  reais for the ticket.

## Output

Print a single integer: the maximum total value, in reais, that Viação Mandacaru can earn.

## Examples

| Sample input 1                                  | Sample output 1 |
|-------------------------------------------------|-----------------|
| <pre>5 5 2 1 3 8 2 5 10 1 2 6 3 5 7 2 4 9</pre> | <pre>31</pre>   |

### Explanation of sample 1

One optimal possibility is to issue the first four requests, earning  $8 + 10 + 6 + 7 = 31$  reais. Passengers getting off in cities 2 and 3 free seats before the new passengers board in those cities.

**Sample input 2**

```
4 6 1
1 4 10
1 2 4
2 3 4
3 6 4
```

**Sample output 2**

```
12
```

**Explanation of sample 2**

Since the bus has only one seat, it is impossible to issue tickets for trips that share any segment. By issuing the last three requests, the same seat can be reused in cities 2 and 3, and the company earns  $4 + 4 + 4 = 12$  reais.