0_Hello World

(30 points)

Introduction

YTP Contest has started!

Let's verify everything first.

Is the internet setting correct?

Is the source code submission working well?

Do you use STDOUT output for program solutions?

If everything is ready, 30 points are yours! Go! Go! Go!

Description

Please write a program to output Hello World!

Input Format

This problem requires no input.

Output Format

[A~Z][a~z], space, and common English punctuation.

Constraints

[A~Z][a~z], space, and exclamation mark "!".

Input Example 1

(no input)

Output Example 1

Hello World!

Example Explanation

Input Example 1 has no input, simply output Hello World!

1_Stickers on Bulletin

(5 points)

Description

Xiao Ming's class moved to a new classroom in a new semester. His classmates purchased some stickers in order to decorate the bulletin board in the back of their classroom. As the class secretary, Xiao Ming came upon an idea that they can use "The class number of their class" stickers to form a rectangle. This way, the bulletin board may be more neat and more spectacular, and this will become a special feature of their class.

However, both the length and height of the bulletin board are limited, so if the length of the rectangle exceed the length of the bulletin board, or if the width of the rectangle exceed the height of the bulletin board, then this rectangle cannot be pasted on the bulletin board. Here, both the length and width of a sticker are 1.

Since given the number of stickers (also their class number), the feasible combinations of length and width of the rectangle may still be so many. Moreover, we also have to take the length and height of the bulletin board into consideration. Thus, Xiao Ming asked you to help him to write a program that can list all feasible rectangles that can be pasted on the bulletin board.

Input Format

The input consists of one line, containing 3 integers: N, L, and H, separated by a space bar.

Here, N denotes the class number, L and H respectively denote the length and height of the bulletin.

Output Format

You should output x lines, where x is the number of types of feasible rectangles.

In the i-th line of your output ($1 \leq i \leq x$), you should print two integers l_i and w_i, separated by a space bar. Here, l_i and w_i respectively denote the length and width of one of the feasible rectangles. Besides, your output should satisfy $l_1 < l_2 < \ldots < l_x$.

Constraints

$1 \leq N \leq 1000$.
$1 \leq L \leq 1000$.
$1 \leq H \leq 1000$.

Input Example 1

```
104 20 20
```

Output Example 1

```
8 13
13 8
```

Input Example 2

```
220 50 15
```

Output Example 2

```
20 11
22 10
44 5
```

Input Example 3

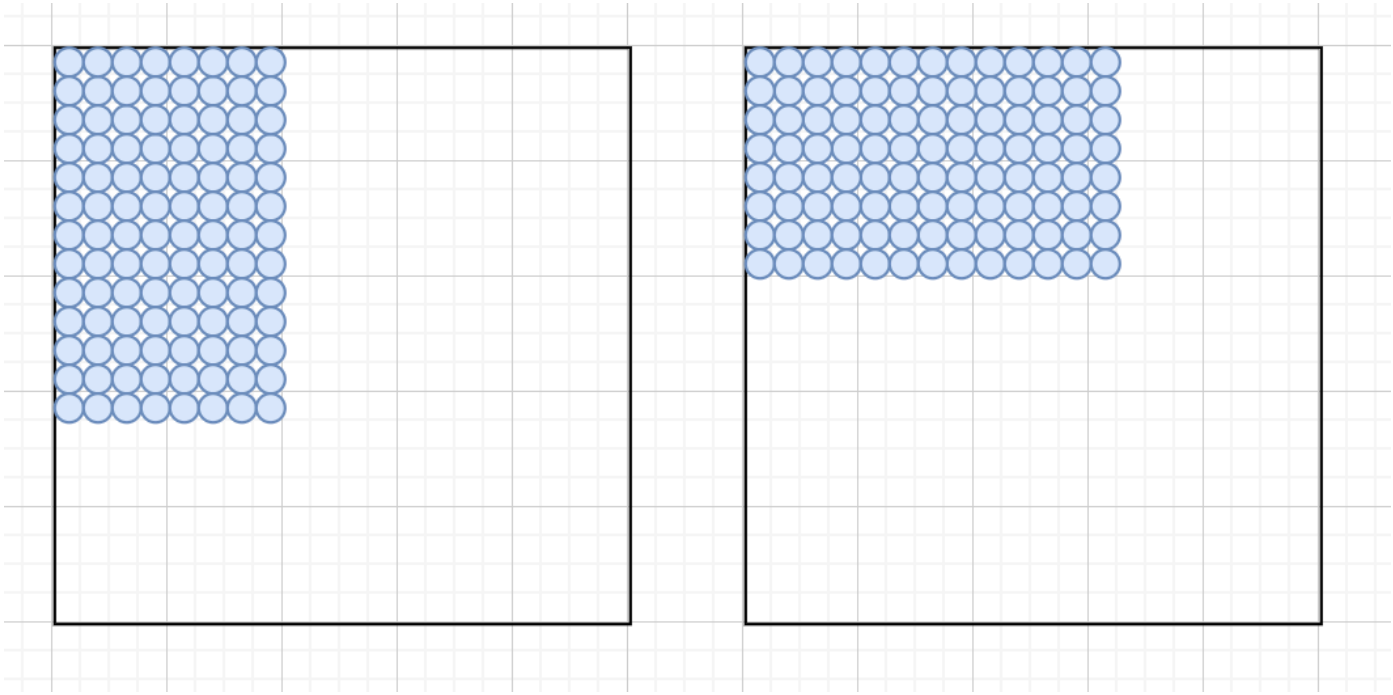
```
327 10 1000
```

Output Example 3

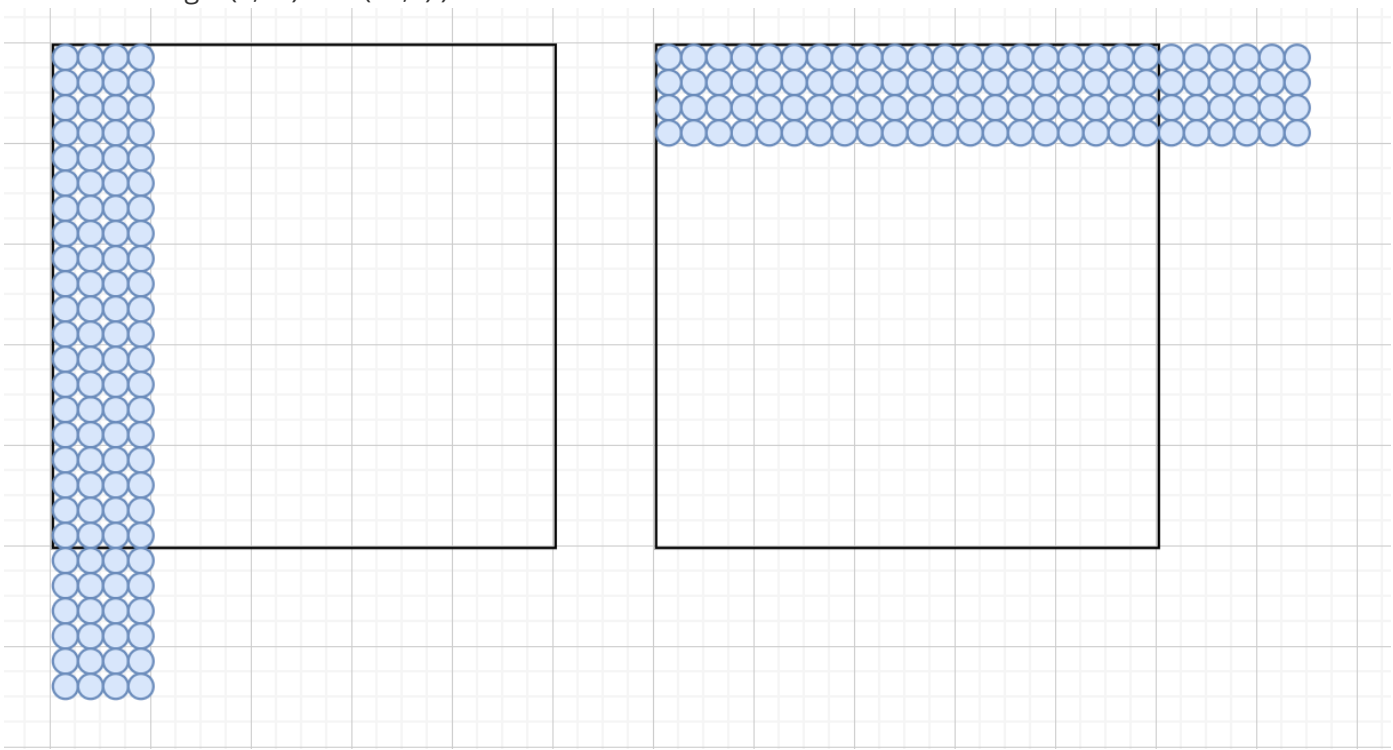
```
1 327
3 109
```

Example Explanation

In example one, their class number is 104, so the length and width of the rectangle can be (1,104), (2, 52), (4, 26), (8, 13), (13, 8), (26, 4), (52, 2), and (104, 1). However, since both the length and height of the bulletin are 20, so only (8, 13) and (13, 8) are feasible.



And, all of the other rectangles are not feasible because they will be out of bounds. (The following picture are the rectangle (4,26) and (26,4).)



In example two, their class number is 220, so the length and width of the rectangle can be (1,220), (2, 110), (4, 55), (5, 44), (10, 22), (11, 20), (20, 11), (22, 10), (44, 5), (55, 4), (110, 2), and (220, 1). However, since length of the bulletin is 50, and the height of it is 15, so only (20, 11), (22, 10), and (44, 5) are feasible.

In example three, their class number is 327, so the length and width of the rectangle can be (1,327), (3, 109), (109, 3), and (327, 1). However, since length of the bulletin is 10, and the height of it is 1000, so only (1, 327) and (3, 109) are feasible.

2_Table Tennis

(10 points)

Problem Description

GrandMaster Cheng, the grandmaster of table tennis, and Noob Cheng, the newbie of table tennis took the same table tennis class this semester. The teacher of this class asks everyone to join a promotion and relegation competition in the first class. The rules are as follows:

There are N tables labeled 1 to N in the court. In the beginning, $2N$ students in the class will be randomly assigned to a table with two students. In a round of the competition, two students assigned to the same table will play a game, the winner in the i-th table will move to the $i-1$-th table for the next round, while the loser will move to the $i+1$-th table for the next round. A bit different from the other tables, the winner in the first table will stay at the same table for the next round. Similarly, the loser in the N-th table will stay at the same table for the next round. It's easy to check after this operation, there should be exactly two students assigned to each table, then the next round begins.

At first, GrandMaster Cheng is assigned to the t_1-th table, while Noob Cheng is assigned to the t_2-th table. GrandMaster Cheng wants to play with Noob Cheng. What is the least number of rounds to achieve GrandMaster Cheng's wish? (Assume that GrandMaster Cheng can control Noob Cheng to win/lose a game, he can also choose to win/lose a game.)

Input format

The only line in the input contains N, t_1, t_2 three positive integers shown as problem description.

N represents the number of tables in the court.

t_1, t_2 represents the starting position (index of the table) of GrandMaster Cheng and Noob Cheng.

Output format

Output an integer in one line, representing the least number of rounds for GrandMaster Cheng to reach Noob Cheng.

Constraints

- $1 \leq N \leq 10^9$
- $1 \leq t_1, t_2 \leq N$

Sample input 1

```
5 3 1
```

Sample output 1

```
1
```

Sample input 2

```
6 3 6
```

Sample output 2

```
2
```

Sample input 3

```
4 2 2
```

Sample output 3

```
0
```

Example Explanation

- Case 1. If Professor Cheng wins and Noob Cheng loses in the first round, both of them will appear at the 2nd table.
- Case 2. If Professor Cheng loses and Noob Cheng loses in the first round, Professor Cheng loses and Noob Cheng wins in the first round, both of them will appear at the 5th table.
- Case 3. Both of them are at the 2nd table before the 1st round.

3_Friendship

(10 points)

Description

David is Leo's best friend and David's birthday is coming. Leo knows that David likes math very much, so Leo decided to prepare a pair of rings and engrave a pair of "amicable numbers" on them respectively. One for David, one for himself, represents their friendship.

What are the "amicable numbers"? Amicable numbers are two different natural numbers related in such a way that the sum of the proper divisors (A positive divisor of n which is different from n is a proper divisor of n) of each is equal to the other number. For example, the proper divisors of 220 are $1, 2, 4, 5, 10, 11, 20, 22, 44, 55$ and 110, of which the sum is 284; and the proper divisors of 284 are $1, 2, 4, 71$ and 142, of which the sum is 220. Hence, $(220, 284)$ is a pair of amicable numbers.

However, Leo wants to find other amicable numbers, then choose the best number to engrave on their rings. So, he hopes you can help him to write a program to find a number's amicable numbers.

Input Format

The only line in the input contains an integer N shown as problem description.
N represents the number of tables in the court.

Output Format

If N has an amicable number, print the number in a line. Otherwise, print "Not found" (without double quotation mark) in a line.

Constraints

- $1 \leq N \leq 10^{12}$

Input Example 1

```
220
```

Output Example 1

```
284
```

Input Example 2

```
6
```

Output Example 2

```
Not found
```

Input Example 3

```
12
```

Output Example 3

```
Not found
```

Example Explanation

- In example 1, the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, and 110, of which the sum is 284; and the proper divisors of 284 are 1, 2, 4, 71, and 142, of which the sum is 220. Hence, they are a pair of amicable numbers.
- In example 2, the proper divisors of 6 are 1, 2, and 3, of which the sum is 6, but 6 can't make a friend with itself. Hence 6 doesn't have an amicable number.
- In example 3, the proper divisors of 12 are 1, 2, 3, 4, and 6, of which the sum is 16; but the proper divisors of 16 are 1, 2, 4, 8, of which the sum is 15. Hence 12 doesn't have an amicable number.

4_Multi-level subtractions

(10分)

Description

Let's briefly review the content of middle school mathematics first!

x power n , noted as x^n, which is x multiple itself n times.

Arithmetic progression, for $x_n = x_1 + (n-1) \times d$, d is tolerance.

Equivalence series, for $a_n = a_1 \times r^{(i-1)}$, r is ratio.

In this problem, we will give you an initial value of an arithmetic sequence x_1, a tolerance d and a power p, so that you can get a power sequence of an arithmetic sequence through these three numbers, and also it is $x_1^p, (x_1+d)^p, \ldots, (x_1+(n-1)\times d)^p$.

Give a practical example! Assuming $x_1 = 1, d = 2, p = 2$, we can first obtain an arithmetic sequence $1, 3, 5, 7, \ldots$ through $x_1 = 1, d = 2$, and then we are correcting each number raised to the power of $p = 2$ will become $1, 9, 25, 49, \ldots$.

Once we have this sequence, we will want to differentiate the sequence several times, that is, subtract adjacent numbers, until all numbers are the same. Taking the above example, we will get $8, 16, 24, \ldots$ after doing one difference of $1, 9, 25, 49, \ldots$, and if we do it again, we can get $8, 8, 8, \ldots$. Since all the numbers are the same, the difference operation is no longer performed!

As we all know that any arithmetic sequence with tolerance and power, a fixed value will be obtained after multiple differential operations. Because Ruby spent a lot of time in the calculation, a high degree of crisis has arisen, so she needs your help to write a program with different inputs. The initial value of the arithmetic sequence is x_1, the tolerance d and the power p, and an integer is output, which represents the fixed value after completing several differences.

Input Format

The input contains three positive integers in one line, x_1, d, p are separated by commas, which represent the initial value, tolerance and power of the arithmetic sequence, respectively.

Output Format

Output an integer on one line, representing the fixed value after performing several differences.

Constraints

- $1 \leq$ x_1, d, p ≤ 6
- We guarantee that the fixed value after the difference is an integer between 1 and $2^{31} - 1 = 2147483647$.

Example 1 Input

```
1,2,2
```

Example 1 Output

```
8
```

Example 2 Input

```
1,2,3
```

Example 2 Output

```
48
```

- Description:

1,2,3 means the series starts at 1 and increase by 2 and do a power 3 for each value. So, the series is $1^3,\ 3^3, 5^3, 7^3, 9^3$, ... => 1, 27, 125, 343, 729, ... etc. Thru the 1^{st} round subtraction, the series becomes 26, 98, 218,386, and so on. After 2^{nd} round subtraction, the series becomes 72,120,168, and so on. With 3^{rd} round subtraction, the series becomes 48, 48, 48, and so on. So, it program returns 48 as result.

Example 3 Input

```
6,6,6
```

Example 3 Output

```
33592320
```

5_Price of chosen numbers

(15分)

Description

When issue a phone or plate, the numbers could be chosen with one preferred with various cost. So, the system manages those numbers must set common, special, or continuous numbers to various unit costs at first.

To simplify the calculation, we group the numbers (for phone or plate) to $1, 2, 5, 10, 20$, and/or 50 unit(s) of cost as following:

Pattern of the numbers	Sample	Unit
Non-special numbers	135246, 505152	1
3 same numbers or continuous numbers in a roll.	555345, 237659	2
2 sets of “same” numbers	111222, 800800 333777、955955	5
4 same numbers or continuous numbers in a roll.	123400, 123333	10
5 same numbers or continuous numbers in a roll.	644444, 900000	20
6 same numbers or continuous numbers in a roll.	456789, 888888	50

PS: “continous numbers” means ascending or descending continuous digits in $0-9$ order (non-circle). Such as, 012, and 987 are continuous numbers. Also, 901, 468 are not continuous numbers.

Please write a program to calculate multiple 6-digits records and then sum up the unit of costs.

Input Format

- An utf-8 formatted text file; every line contains a 6-digits number.
- Input ends at the line does not have a 6-digits number.

Output Format

Output an integer on one line, representing the total sum of units, with a value less than 50001.

Constraints

The program ends the calculation and print the sum If the line does not have a 6-digits number. It is less and equal to 1000 lines.

Example 1 Input

```
888888
135246
147258
555345
end
```

Example 1 Output

```
54
```

Example 2 Input

```
237659
111222
800800
123400
123333
644444
end
```

Example 2 Output

```
52
```

Example 3 Input

```
900000
456789
909090
901357
000000
end
```

Example 3 Output

```
122
```

Example Explaination

- Example 1：50+1+1+2=54

6_SING TONG YANG

(15 points)

Description

The hot summer inspires Tommy and Yang, the owners of SING TONG YANG, which is a food corporation famous for beef jerky, to invest in sunglasses business. With limited experience for the clothing industry, they try to figure out a marketing strategy to promote their new products before they hit the market.
Tommy proposes to hold a Big Eater Game which combines their popular flash meat buns with sunglasses to draw attention. Game players should consume at least three meat buns successively within the given D minutes, and each individual meat bun should be finished in every single minute. In other words, game players can choose to eat or take a break in every single minute. If they accomplish the mission, they will win a pair of sunglasses.
Now Tommy and Yang want you to help them figure out how many different ways can game players win sunglasses within D minutes?

Input Format

The input contains only one number D, representing the total minute of the activity.

Output Format

The output contains only one number Y MOD $10^9 + 7$, representing the number of possible ways to get a sunglasses within D minute.

Constraints

- $0 \leq D \leq 100000$

Input Example 1

```
1
```

Output Example 1

```
0
```

Input Example 2

```
3
```

Output Example 2

```
1
```

Input Example 3

```
4
```

Output Example 3

```
3
```

Example Explanation

- Example One: It is impossible to eat three bright meat bun continuously.
- Example Two: (eat , eat , eat) => One option
- Example Three: (eat , eat , eat , rest) , (rest , eat , eat , eat) , (eat , eat , eat , eat) => Three options

7_Quick Waiting Line

(10 points /10 points)

Problme Description

Held by students from the Department of Computer Science, T Super University, the oldest and most professional programming competition camp YTPCamp is about to start registration! As a program camp, the registration method of YTPCamp is of course... on-site registration! All doors of the Department of Computer Science - the DT Building will be used as the registration window at that time, and the staff will collect and review the registration form.

The review of each page of form takes 1 minute. When the number of people increases, it may take several hours to line up. The most frustrating thing is that there is only 1 page of registration form to review, but there is a person in front with 100 pages of form waiting to be reviewed.

As the general manager of YTPCamp, Casper is very nervous, afraid that the students would get angry after waiting too long. Therefore, he pre-sets some of the gates of the DT Building as express gates, so that students with fewer pages of registration form can use the express gates for priority review and reduce the total waiting time.

There are N gates for queuing in the Department of Computer Science. Among them, there are M gates that can only be queued by students whose number of pages does not exceed K. The total waiting time of students is defined as the sum of waiting time of each student. The waiting time of a student is the sum of the registration form pages of all the people in front of him.

A few hours before the official registration began, students who could not wait to register had already arrived in front of the DT Building one after another. Smart students have the ability to quickly identify the queue with the minimum waiting time, so they will always choose the door with the least queuing time among the available doors (if there are two doors that require the same queuing time, he will select the express gate first).

Casper already knows the number of pages of registration form a_i for each student who came to the department in chronological order, can you help Casper calculate what the total waiting time will be for these students, so that he can buy them some drinks in compensation?

Input Format

Three integers N, M, K are in the first line. There are N gates, M of them are express gates, and students whose number of pages do not exceed K can be lined in front of the express gate.

The second line has an integer A, representing a total of A students waiting at the gate.

The third line has A integers, and the i th integer a_i represents the i student who arrived in the department has a_i number of pages.

Output Format

Output an integer, representing the total waiting time for students.

Constraints

- $2 \leq N \leq 10000$
- $1 \leq M < N$
- $1 \leq K \leq 1000$
- $1 \leq A \leq 100000$
- $1 \leq a_i \leq 1000$

Subtasks

- Subtask 1 satisfies K=1000 (10 points).
- There is no additional constraint to Subtask 2 (10 points).

Input Example 1

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

Output Example 1

```
8
```

Input Example 2

```
5 1 5
10
10 9 8 7 6 5 4 3 2 1
```

Output Example 2

```
38
```

Input Example 3

```
5 3 1000
5
4 4 3 1 2
```

Output Example 3

```
0
```

Example Explanation

- In example 1, the 1st person goes to the normal gate and the waiting time is 0.

  The 2nd person goes to the second normal gate and the wait time is 0.

  The 3rd person goes to the express gate and the wait time is 0.

  The 4th person is waiting behind the 2nd person and the waiting time is 3.

  The 5th person is waiting behind the 1st person and the wait time is 4.

  The 6th person is waiting behind the 3rd person and the waiting time is 1.

  Therefore, the total waiting time is 8.

- In example 2, there is no one in front of the 1st, 2nd, 3rd, 4th, and 6th people, and the waiting time is 0. (6th person is at the express gate)

  The 5th person is behind the 4th person and the wait time is 7.

  The 7th person is behind the 5th person and the wait time is 5.

  The 8th person is behind the 3rd person and the wait time is 8.

  The 9th person is behind the 7th person (express gate first), and the waiting time is 9.

  The 10th person is behind the 2nd person and the wait time is 9.

  Therefore, the total wait time is 38.

- In example 3, there is no one in front of everyone, and the total waiting time is 0.

8_Maximum Sum

(20 points)

Time Limit: 1 second
Memory Limit: 512MB

Description

In many problems, the answers may be so large that you need to modulo some number P before you print the answers.
This problem is no exception.
You are given N positive integers $A_1, A_2, \cdots, A_N$.
You can select some of them. Let the sum of the chosen elements be X.
Find the maximum value of $X \mod (10^{17}+3)$ you can achieve.

```
A={
  43257517791815812,
  7158485778091600,
  22932684354088977,
  26557122523572685,
  94189552430929
}

A1+A2+A3+A4=99905810447569074

(A1+A2+A3+A4+A5) mod 100000000000000003=0
```

Input Format

The first line of the input contains an integer N, representing the number of positive integers.
The second line of the input contains N positive integers $A_1, A_2, \cdots, A_N$.

Output Format

You should print a positive integer representing the maximum value $X \mod (10^{17}+3)$ you can achieve.

Constraints

- $1 \leq N \leq 35$
- $1 \leq A_i \leq 10^{17}+3$

Input Example 1

```
1
1
```

Output Example 1

```
1
```

Input Example 2

```
2
50000000000000002 50000000000000003
```

Output Example 2

```
50000000000000003
```

Input Example 3

```
5
43257517791815812 7158485778091600 22932684354088977 26557122523572685 94189552430929
```

Output Example 3

```
99905810447569074
```

Example Explanation

- In example one, the only possible value of X is 1, so the maximum value of $X \mod (10^{17}+3)$ is also 1.
- In example two, the possible values of X are $(50000000000000002 + 50000000000000003) = 10^{17}+5$, 50000000000000002, 50000000000000003.
  The maximum value of $X \mod (10^{17}+3)$ is 50000000000000003.
- In example three, you should choose A_1, A_2, A_3, A_4 and their sum is 99905810447569074.

9_Mai The MTB Guy

(20 points)

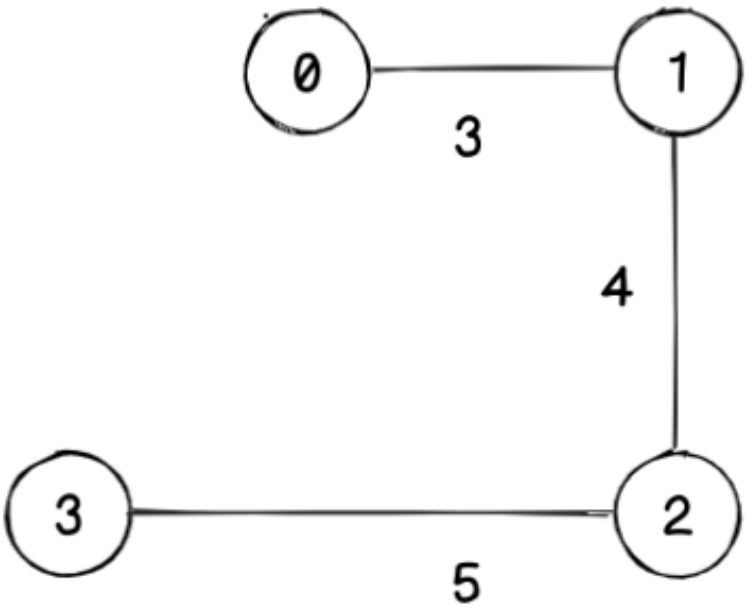
Time limit: 3 seconds

Memory limit: 512 MB

Description

William Mai, also known as The Mountain Bike (MTB) Mai, is a guy who rides his mountain bike every day. As a senior member of Crazy Cycling Club, he wants to popularize mountain bike sport by doing something crazy: finding the craziest MTB route in this country, and then he will finish it.
Mai the MTB Guy lives in a country that all peaks are connected, that is, for any two peaks, there must be a route connecting them. Also, suppose there are E roads connecting the peaks, there will be exactly $E+1$ peaks.

To determine the craziness of an MTB route, Mai first takes two peaks in the country, and then find the distance between two peaks. The distance between two peaks is determined by the shortest route between two peaks. For example, look at the following graph:



In the above graph, the circles with numbers inside them represent the peaks and their numbering, and the black lines connecting the circles represent the lengths of the roads.

The shortest route between peak 0 and peak 3 is $(0 \to 1 \to 2 \to 3)$, so the distance between the two peaks is 12. Moreover, such a route is also called a **crazy route**, since its length equals to the distance between its two endpoints.

It's rumored that longest crazy route has a magic power, and its distance is called the **magic crazy distance**.

Please help Mai the MTB Guy find the magic crazy distance of the country, so he can focus on preparing for the trip.

Input Format

The first line contains an integer E, indicating the number of roads of this country. Each of the following E lines contains 3 integers a_i, b_i, c_i representing a road. The integers a_i, b_i indicate the 2 endpoints of the road, while c_i indicates the length of the road.

Output Format

Output the magic crazy distance of the country and end it with a new line.

Constraints

- $1 \leq E \leq 1000000$
- $0 \leq a_i, b_i \leq E$ and $0 < c_i \leq 2^{31} - 1$ for each $i = 1, 2, \ldots, E$

Input Example 1

```
3
0 1 3
1 2 4
2 3 5
```

Output Example 1

```
12
```

Input Example 2

```
5
0 1 3
0 2 4
0 3 5
0 4 1
0 5 8
```

Output Example 2

```
13
```

Input Example 3

```
5
0 1 10
1 2 10
1 3 10
1 4 10
4 5 10
```

Output Example 3

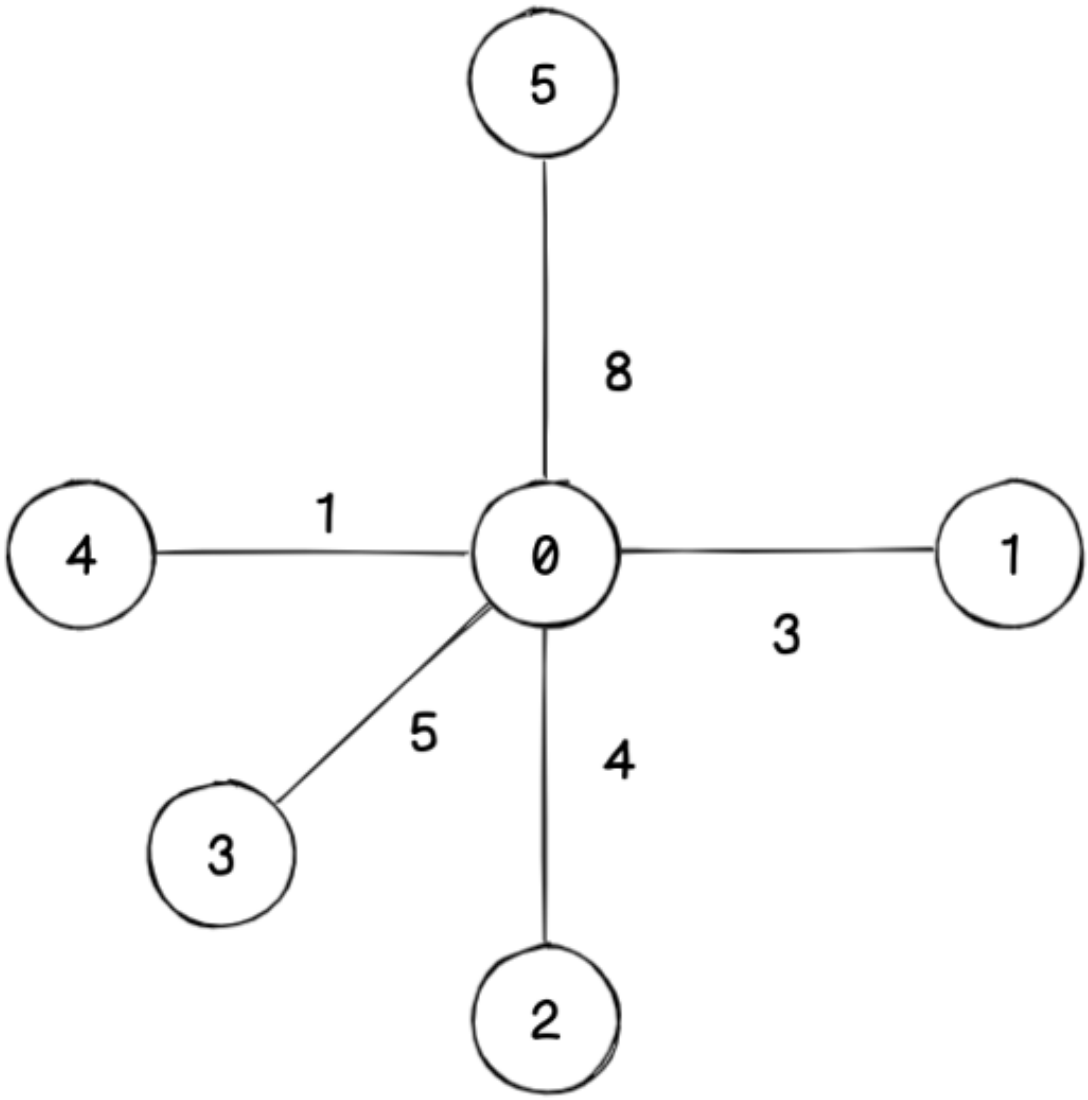
```
30
```

Example Explanation

- Example 1

  This is explained in the problem description.

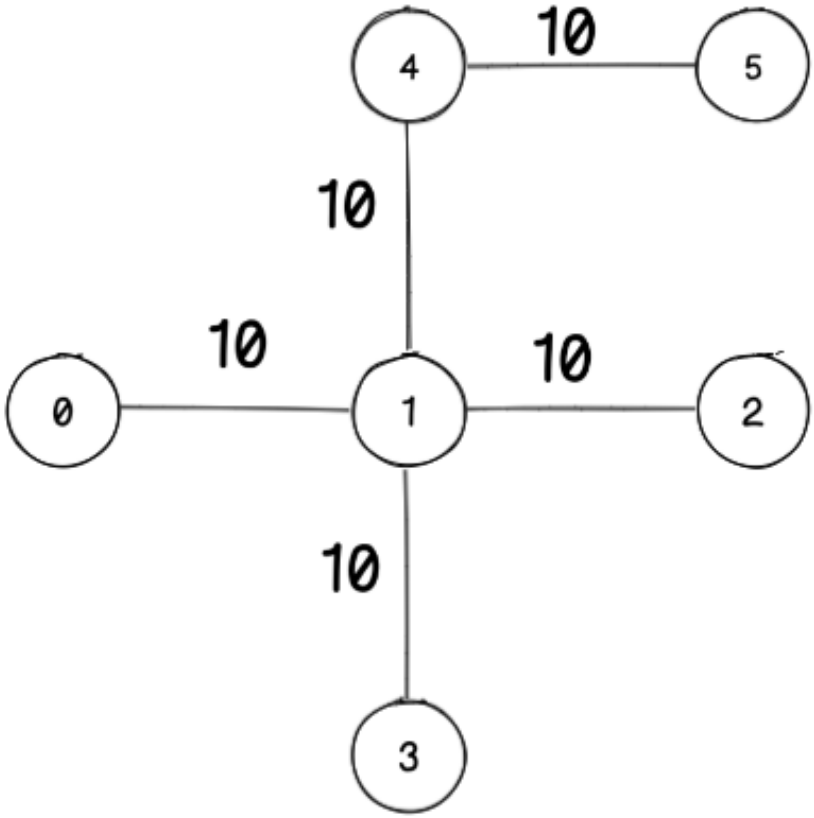
- Example 2



The longest route is $(3 \to 0 \to 5)$.

Hence, this country's magic crazy distance is 13.

- Example 3



From peak 0 to peak 5, the distance is 30.

10_Frog Crossing River

(25 points)

Time Limit: 3 second
Memory Limit: 512MB

Description

A frog is now at one bank of a river, and he wants to step on the rocks revealing above the surface to cross the river.

Here, we can map the positions of the stones to lattice points on a 2-D plane. Given a rock revealing above the surface, we can use (m, n) to denote its position, where $1 \leq m \leq (width\_of\_the\_river)$, $1 \leq n \leq (length\_of\_the\_river)$.

How the frog crosses the river is that he will choose a rock among all rocks satisfying $m = 1$. Suppose the rock is at $(1, n_1)$, then he will jump from the bank to $(1, n_1)$. Then, he chooses a rock among all rocks satisfying $m = 2$, suppose the rock is at $(2, n_2)$, then he will jump from $(1, n_1)$ to $(2, n_2)$... So, if the width of river is W, then at the last, he will jump from $(W - 1, n_{W-1})$ to (W, n_W), and jump from (W, n_W) to the opposite bank.

However, different jumping distances may consume different amounts of energies. Note that the frog won't consume energy when jumping from the river bank to $(1, n_1)$ nor jumping from (W, n_W) to the opposite bank of the river. However, jumping from (i, n_i) to $(i + 1, n_{i+1})$ will cost the frog $|n_i - n_{i+1}|$ energies.

Given the length and width of the river and positions of all rocks revealing above the surface of the river, please help the frog to calculate the minimum amount he will consume to cross the river.

Input Format

- $L\ W$
- $N_1\ r_{(1,1)}\ r_{(1,2)}$ ... $r_{(1,N_1)}$
- $N_2\ r_{(2,1)}\ r_{(2,2)}$ ... $r_{(2,N_2)}$
- ...
- $N_W\ r_{(W,1)}\ r_{(W,2)}$ ... $r_{(W,N_W)}$

The first line of the input consists of two integers, L and W, which respectively denotes the length and width of the river.

In the next W lines, the i-th line $(1 \leq i \leq W)$ contains a positive integer N_i followed by N_i integers: $r_{(i,1)}$, $r_{(i,2)}$, ..., and $r_{(i,N_i)}$. Which means that there are rocks revealing above the surface of river at $(i, r_{(i,1)})$, $(i, r_{(i,2)})$, ..., and $(i, r_{(i,N_i)})$.

Here, $1 \leq r_{(i,1)} < r_{(i,2)} < \ldots < r_{(i,N_i)} \leq L$ for all i.

Output Format

Your output consists of only one integer, which is the minimum amount of energies that the frog will consume. Remember to add a '\n' at the end of that integer.

Constraints

- $1 \leq L \leq 100000$.
- $1 \leq W \leq 100$.
- $\sum_{i=1}^{W} N_i \leq 1000000$
- $1 \leq N_i \leq L$, $\forall 1 \leq i \leq W$

Input Example 1

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

Output Example 1

```
2
```

Input Example 2

```
5 3
1 5
3 1 2 3
1 5
```

Output Example 2

```
4
```

Input Example 3

```
10 4
3 1 6 9
2 1 8
3 3 4 5
4 1 4 5 7
```

Output Example 3

```
3
```

Example Explanation

In example one, if the frog jumps on the rocks at $(1, 4)$, $(2, 3)$, and $(3, 2)$, then he only has to consume $|4 - 3| + |3 - 2| = 2$ energies.

In example two, if the frog jumps on the rocks at $(1, 5)$, $(2, 3)$, and $(3, 5)$, then he only has to consume $|5 - 3| + |3 - 5| = 4$ energies.

In example three, if the frog jumps on the rocks at $(1, 1)$, $(2, 1)$, $(3, 3)$, and $(4, 4)$, then he only has to consume $|1 - 1| + |1 - 3| + |3 - 4| = 3$ energies.