A - Frog 1

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N stones, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), the height of Stone i is h_i.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N:

- If the frog is currently on Stone i, jump to Stone $i + 1$ or Stone $i + 2$. Here, a cost of $|h_i - h_j|$ is incurred, where j is the stone to land on.

Find the minimum possible total cost incurred before the frog reaches Stone N.

Constraints

- All values in input are integers.
- $2 \leq N \leq 10^5$
- $1 \leq h_i \leq 10^4$

Input

Input is given from Standard Input in the following format:

$$N$$
$$h_1 \quad h_2 \quad \ldots \quad h_N$$

Output

Print the minimum possible total cost incurred.

Sample Input 1

```
4
10 30 40 20
```

Sample Output 1

```
30
```

If we follow the path $1 \to 2 \to 4$, the total cost incurred would be $|10 - 30| + |30 - 20| = 30$.

Sample Input 2

```
2
10 10
```

Sample Output 2

```
0
```

If we follow the path $1 \to 2$, the total cost incurred would be $|10 - 10| = 0$.

Sample Input 3

```
6
30 10 60 10 60 50
```

Sample Output 3

```
40
```

If we follow the path $1 \to 3 \to 5 \to 6$, the total cost incurred would be $|30 - 60| + |60 - 60| + |60 - 50| = 40$.

B - Frog 2

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N stones, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), the height of Stone i is h_i.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N:

- If the frog is currently on Stone i, jump to one of the following: Stone $i + 1, i + 2, \ldots, i + K$. Here, a cost of $|h_i - h_j|$ is incurred, where j is the stone to land on.

Find the minimum possible total cost incurred before the frog reaches Stone N.

Constraints

- All values in input are integers.
- $2 \leq N \leq 10^5$
- $1 \leq K \leq 100$
- $1 \leq h_i \leq 10^4$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad K \\
h_1 \quad h_2 \quad \ldots \quad h_N
\end{array}
$$

Output

Print the minimum possible total cost incurred.

Sample Input 1

```
5 3
10 30 40 50 20
```

Sample Output 1

```
30
```

If we follow the path $1 \to 2 \to 5$, the total cost incurred would be $|10 - 30| + |30 - 20| = 30$.

Sample Input 2

```
3 1
10 20 10
```

Sample Output 2

```
20
```

If we follow the path $1 \to 2 \to 3$, the total cost incurred would be $|10 - 20| + |20 - 10| = 20$.

Sample Input 3

```
2 100
10 10
```

Sample Output 3

```
0
```

If we follow the path $1 \to 2$, the total cost incurred would be $|10 - 10| = 0$.

Sample Input 4

```
10 4
40 10 20 70 80 10 20 70 80 60
```

Sample Output 4

```
40
```

If we follow the path $1 \to 4 \to 8 \to 10$, the total cost incurred would be $|40 - 70| + |70 - 70| + |70 - 60| = 40$.

C - Vacation

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Taro's summer vacation starts tomorrow, and he has decided to make plans for it now.

The vacation consists of N days. For each i ($1 \leq i \leq N$), Taro will choose one of the following activities and do it on the i-th day:

- A: Swim in the sea. Gain a_i points of happiness.
- B: Catch bugs in the mountains. Gain b_i points of happiness.
- C: Do homework at home. Gain c_i points of happiness.

As Taro gets bored easily, he cannot do the same activities for two or more consecutive days.

Find the maximum possible total points of happiness that Taro gains.

Constraints

- All values in input are integers.
- $1 \leq N \leq 10^5$
- $1 \leq a_i, b_i, c_i \leq 10^4$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
a_1 \quad b_1 \quad c_1 \\
a_2 \quad b_2 \quad c_2 \\
: \\
a_N \quad b_N \quad c_N
\end{array}
$$

Output

Print the maximum possible total points of happiness that Taro gains.

Sample Input 1

```
3
10 40 70
20 50 80
30 60 90
```

Sample Output 1

```
210
```

If Taro does activities in the order C, B, C, he will gain $70 + 50 + 90 = 210$ points of happiness.

Sample Input 2

```
1
100 10 1
```

Sample Output 2

```
100
```

Sample Input 3

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

Sample Output 3

```
46
```

Taro should do activities in the order C, A, B, A, C, B, A.

D - Knapsack 1

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N items, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), Item i has a weight of w_i and a value of v_i.

Taro has decided to choose some of the N items and carry them home in a knapsack. The capacity of the knapsack is W, which means that the sum of the weights of items taken must be at most W.

Find the maximum possible sum of the values of items that Taro takes home.

Constraints

- All values in input are integers.
- $1 \leq N \leq 100$
- $1 \leq W \leq 10^5$
- $1 \leq w_i \leq W$
- $1 \leq v_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad W \\
w_1 \quad v_1 \\
w_2 \quad v_2 \\
: \\
w_N \quad v_N
\end{array}
$$

Output

Print the maximum possible sum of the values of items that Taro takes home.

Sample Input 1

```
3 8
3 30
4 50
5 60
```

Sample Output 1

```
90
```

Items 1 and 3 should be taken. Then, the sum of the weights is $3+5=8$, and the sum of the values is $30+60=90$.

Sample Input 2

```
5 5
1 1000000000
1 1000000000
1 1000000000
1 1000000000
1 1000000000
```

Sample Output 2

```
5000000000
```

The answer may not fit into a 32-bit integer type.

Sample Input 3

```
6 15
6 5
5 6
6 4
6 6
3 5
7 2
```

Sample Output 3

```
17
```

Items $2, 4$ and 5 should be taken. Then, the sum of the weights is $5 + 6 + 3 = 14$, and the sum of the values is $6 + 6 + 5 = 17$.

E - Knapsack 2

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N items, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), Item i has a weight of w_i and a value of v_i.

Taro has decided to choose some of the N items and carry them home in a knapsack. The capacity of the knapsack is W, which means that the sum of the weights of items taken must be at most W.

Find the maximum possible sum of the values of items that Taro takes home.

Constraints

- All values in input are integers.
- $1 \leq N \leq 100$
- $1 \leq W \leq 10^9$
- $1 \leq w_i \leq W$
- $1 \leq v_i \leq 10^3$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad W \\
w_1 \quad v_1 \\
w_2 \quad v_2 \\
: \\
w_N \quad v_N
\end{array}
$$

Output

Print the maximum possible sum of the values of items that Taro takes home.

Sample Input 1

```
3 8
3 30
4 50
5 60
```

Sample Output 1

```
90
```

Items 1 and 3 should be taken. Then, the sum of the weights is $3+5=8$, and the sum of the values is $30+60=90$.

Sample Input 2

```
1 1000000000
1000000000 10
```

Sample Output 2

```
10
```

Sample Input 3

```
6 15
6 5
5 6
6 4
6 6
3 5
7 2
```

Sample Output 3

```
17
```

Items $2, 4$ and 5 should be taken. Then, the sum of the weights is $5+6+3=14$, and the sum of the values is $6+6+5=17$.

F - LCS

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given strings s and t. Find one longest string that is a subsequence of both s and t.

Notes

A *subsequence* of a string x is the string obtained by removing zero or more characters from x and concatenating the remaining characters without changing the order.

Constraints

- s and t are strings consisting of lowercase English letters.
- $1 \leq |s|, |t| \leq 3000$

Input

Input is given from Standard Input in the following format:

s
t

Output

Print one longest string that is a subsequence of both s and t. If there are multiple such strings, any of them will be accepted.

Sample Input 1

```
axyb
abyxb
```

Sample Output 1

```
axb
```

The answer is `axb` or `ayb`; either will be accepted.

Sample Input 2

```
aa
xayaz
```

Sample Output 2

```
aa
```

Sample Input 3

```
a
z
```

Sample Output 3

```

```

The answer is (an empty string).

Sample Input 4

```
abracadabra
avadakedavra
```

Sample Output 4

```
aaadara
```

G - Longest Path

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a directed graph G with N vertices and M edges. The vertices are numbered $1, 2, \ldots, N$, and for each i ($1 \leq i \leq M$), the i-th directed edge goes from Vertex x_i to y_i. G **does not contain directed cycles**.

Find the length of the longest directed path in G. Here, the length of a directed path is the number of edges in it.

Constraints

- All values in input are integers.
- $2 \leq N \leq 10^5$
- $1 \leq M \leq 10^5$
- $1 \leq x_i, y_i \leq N$
- All pairs (x_i, y_i) are distinct.
- G **does not contain directed cycles**.

Input

Input is given from Standard Input in the following format:

$$\begin{array}{ll} N & M \\ x_1 & y_1 \\ x_2 & y_2 \\ : & \\ x_M & y_M \end{array}$$

Output

Print the length of the longest directed path in G.

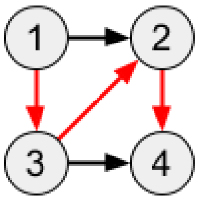
Sample Input 1

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

Sample Output 1

```
3
```

The red directed path in the following figure is the longest:



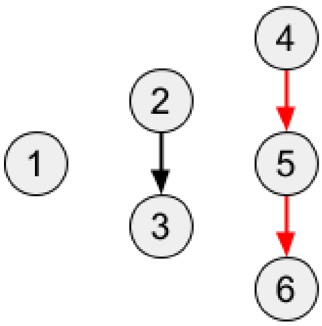
Sample Input 2

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

Sample Output 2

```
2
```

The red directed path in the following figure is the longest:



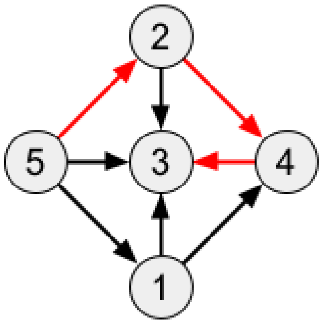
Sample Input 3

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

Sample Output 3

```
3
```

The red directed path in the following figure is one of the longest:



H - Grid 1

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a grid with H horizontal rows and W vertical columns. Let (i, j) denote the square at the i-th row from the top and the j-th column from the left.

For each i and j ($1 \leq i \leq H, 1 \leq j \leq W$), Square (i, j) is described by a character $a_{i,j}$. If $a_{i,j}$ is `.`, Square (i, j) is an empty square; if $a_{i,j}$ is `#`, Square (i, j) is a wall square. It is guaranteed that Squares $(1, 1)$ and (H, W) are empty squares.

Taro will start from Square $(1, 1)$ and reach (H, W) by repeatedly moving right or down to an adjacent empty square.

Find the number of Taro's paths from Square $(1, 1)$ to (H, W). As the answer can be extremely large, find the count modulo $10^9 + 7$.

Constraints

- H and W are integers.
- $2 \leq H, W \leq 1000$
- $a_{i,j}$ is `.` or `#`.
- Squares $(1, 1)$ and (H, W) are empty squares.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
H \quad W \\
a_{1,1} \ldots a_{1,W} \\
\vdots \\
a_{H,1} \ldots a_{H,W}
\end{array}
$$

Output

Print the number of Taro's paths from Square $(1, 1)$ to (H, W), modulo $10^9 + 7$.

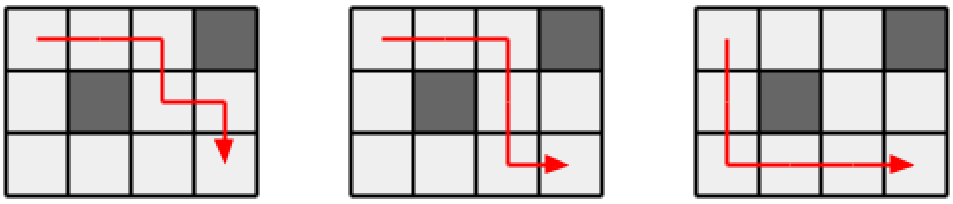
Sample Input 1

```
3 4
...#
.#..
....
```

Sample Output 1

```
3
```

There are three paths as follows:



Sample Input 2

```
5 2
..
#.
..
.#
..
```

Sample Output 2

```
0
```

There may be no paths.

Sample Input 3

```
5 5
..#..
.....
#...#
.....
..#..
```

Sample Output 3

```
24
```

Sample Input 4

```
20 20
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
```

Sample Output 4

```
345263555
```

Be sure to print the count modulo $10^9 + 7$.

I - Coins

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Let N be a positive odd number.

There are N coins, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), when Coin i is tossed, it comes up heads with probability p_i and tails with probability $1 - p_i$.

Taro has tossed all the N coins. Find the probability of having more heads than tails.

Constraints

- N is an odd number.
- $1 \leq N \leq 2999$
- p_i is a real number and has two decimal places.
- $0 < p_i < 1$

Input

Input is given from Standard Input in the following format:

N
$p_1 \quad p_2 \quad \ldots \quad p_N$

Output

Print the probability of having more heads than tails. The output is considered correct when the absolute error is not greater than 10^{-9}.

Sample Input 1

```
3
0.30 0.60 0.80
```

Sample Output 1

```
0.612
```

The probability of each case where we have more heads than tails is as follows:

- The probability of having $(Coin1, Coin2, Coin3) = (Head, Head, Head)$ is $0.3 \times 0.6 \times 0.8 = 0.144$;
- The probability of having $(Coin1, Coin2, Coin3) = (Tail, Head, Head)$ is $0.7 \times 0.6 \times 0.8 = 0.336$;
- The probability of having $(Coin1, Coin2, Coin3) = (Head, Tail, Head)$ is $0.3 \times 0.4 \times 0.8 = 0.096$;
- The probability of having $(Coin1, Coin2, Coin3) = (Head, Head, Tail)$ is $0.3 \times 0.6 \times 0.2 = 0.036$.

Thus, the probability of having more heads than tails is $0.144 + 0.336 + 0.096 + 0.036 = 0.612$.

Sample Input 2

```
1
0.50
```

Sample Output 2

```
0.5
```

Outputs such as `0.500`, `0.500000001` and `0.499999999` are also considered correct.

Sample Input 3

```
5
0.42 0.01 0.42 0.99 0.42
```

Sample Output 3

```
0.3821815872
```

J - Sushi

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N dishes, numbered $1, 2, \ldots, N$. Initially, for each i ($1 \leq i \leq N$), Dish i has a_i ($1 \leq a_i \leq 3$) pieces of sushi on it.

Taro will perform the following operation repeatedly until all the pieces of sushi are eaten:

- Roll a die that shows the numbers $1, 2, \ldots, N$ with equal probabilities, and let i be the outcome. If there are some pieces of sushi on Dish i, eat one of them; if there is none, do nothing.

Find the expected number of times the operation is performed before all the pieces of sushi are eaten.

Constraints

- All values in input are integers.
- $1 \leq N \leq 300$
- $1 \leq a_i \leq 3$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
a_1 \ a_2 \ \ldots \ a_N
\end{array}
$$

Output

Print the expected number of times the operation is performed before all the pieces of sushi are eaten. The output is considered correct when the relative difference is not greater than 10^{-9}.

Sample Input 1

```
3
1 1 1
```

Sample Output 1

```
5.5
```

The expected number of operations before the first piece of sushi is eaten, is 1. After that, the expected number of operations before the second sushi is eaten, is 1.5. After that, the expected number of operations before the third sushi is eaten, is 3. Thus, the expected total number of operations is $1 + 1.5 + 3 = 5.5$.

Sample Input 2

```
1
3
```

Sample Output 2

```
3
```

Outputs such as `3.00`, `3.000000003` and `2.999999997` will also be accepted.

Sample Input 3

```
2
1 2
```

Sample Output 3

```
4.5
```

Sample Input 4

```
10
1 3 2 3 3 2 3 2 1 3
```

Sample Output 4

```
54.48064457488221
```

K - Stones

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a set $A = \{a_1, a_2, \ldots, a_N\}$ consisting of N positive integers. Taro and Jiro will play the following game against each other.

Initially, we have a pile consisting of K stones. The two players perform the following operation alternately, starting from Taro:

- Choose an element x in A, and remove exactly x stones from the pile.

A player loses when he becomes unable to play. Assuming that both players play optimally, determine the winner.

Constraints

- All values in input are integers.
- $1 \leq N \leq 100$
- $1 \leq K \leq 10^5$
- $1 \leq a_1 < a_2 < \cdots < a_N \leq K$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad K \\
a_1 \quad a_2 \quad \ldots \quad a_N
\end{array}
$$

Output

If Taro will win, print `First`; if Jiro will win, print `Second`.

Sample Input 1

```
2 4
2 3
```

Sample Output 1

```
First
```

If Taro removes three stones, Jiro cannot make a move. Thus, Taro wins.

Sample Input 2

```
2 5
2 3
```

Sample Output 2

```
Second
```

Whatever Taro does in his operation, Jiro wins, as follows:

- If Taro removes two stones, Jiro can remove three stones to make Taro unable to make a move.
- If Taro removes three stones, Jiro can remove two stones to make Taro unable to make a move.

Sample Input 3

```
2 7
2 3
```

Sample Output 3

```
First
```

Taro should remove two stones. Then, whatever Jiro does in his operation, Taro wins, as follows:

- If Jiro removes two stones, Taro can remove three stones to make Jiro unable to make a move.
- If Jiro removes three stones, Taro can remove two stones to make Jiro unable to make a move.

Sample Input 4

```
3 20
1 2 3
```

Sample Output 4

```
Second
```

Sample Input 5

```
3 21
1 2 3
```

Sample Output 5

```
First
```

Sample Input 6

```
1 100000
1
```

Sample Output 6

```
Second
```

L - Deque

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Taro and Jiro will play the following game against each other.

Initially, they are given a sequence $a = (a_1, a_2, \ldots, a_N)$. Until a becomes empty, the two players perform the following operation alternately, starting from Taro:

- Remove the element at the beginning or the end of a. The player earns x points, where x is the removed element.

Let X and Y be Taro's and Jiro's total score at the end of the game, respectively. Taro tries to maximize $X - Y$, while Jiro tries to minimize $X - Y$.

Assuming that the two players play optimally, find the resulting value of $X - Y$.

Constraints

- All values in input are integers.
- $1 \leq N \leq 3000$
- $1 \leq a_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$N$$
$$a_1 \quad a_2 \quad \ldots \quad a_N$$

Output

Print the resulting value of $X - Y$, assuming that the two players play optimally.

Sample Input 1

```
4
10 80 90 30
```

Sample Output 1

```
10
```

The game proceeds as follows when the two players play optimally (the element being removed is written bold):

- Taro: (10, 80, 90, **30**) $\rightarrow$ (10, 80, 90)
- Jiro: (10, 80, **90**) $\rightarrow$ (10, 80)
- Taro: (10, **80**) $\rightarrow$ (10)
- Jiro: (**10**) $\rightarrow$ ()

Here, $X = 30 + 80 = 110$ and $Y = 90 + 10 = 100$.

Sample Input 2

```
3
10 100 10
```

Sample Output 2

```
-80
```

The game proceeds, for example, as follows when the two players play optimally:

- Taro: (**10**, 100, 10) $\rightarrow$ (100, 10)
- Jiro: (**100**, 10) $\rightarrow$ (10)
- Taro: (**10**) $\rightarrow$ ()

Here, $X = 10 + 10 = 20$ and $Y = 100$.

Sample Input 3

```
1
10
```

Sample Output 3

```
10
```

Sample Input 4

```
10
1000000000 1 1000000000 1 1000000000 1 1000000000 1 1000000000 1
```

Sample Output 4

```
4999999995
```

The answer may not fit into a 32-bit integer type.

Sample Input 5

```
6
4 2 9 7 1 5
```

Sample Output 5

```
2
```

The game proceeds, for example, as follows when the two players play optimally:

- Taro: (4, 2, 9, 7, 1, **5**) → (4, 2, 9, 7, 1)
- Jiro: (**4**, 2, 9, 7, 1) → (2, 9, 7, 1)
- Taro: (2, 9, 7, **1**) → (2, 9, 7)
- Jiro: (2, 9, **7**) → (2, 9)
- Taro: (2, **9**) → (2)
- Jiro: (**2**) → ()

Here, $X = 5 + 1 + 9 = 15$ and $Y = 4 + 7 + 2 = 13$.

M - Candies

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N children, numbered $1, 2, \ldots, N$.

They have decided to share K candies among themselves. Here, for each i ($1 \leq i \leq N$), Child i must receive between 0 and a_i candies (inclusive). Also, no candies should be left over.

Find the number of ways for them to share candies, modulo $10^9 + 7$. Here, two ways are said to be different when there exists a child who receives a different number of candies.

Constraints

- All values in input are integers.
- $1 \leq N \leq 100$
- $0 \leq K \leq 10^5$
- $0 \leq a_i \leq K$

Input

Input is given from Standard Input in the following format:

$$N \quad K$$
$$a_1 \quad a_2 \quad \ldots \quad a_N$$

Output

Print the number of ways for the children to share candies, modulo $10^9 + 7$.

Sample Input 1

```
3 4
1 2 3
```

Sample Output 1

```
5
```

There are five ways for the children to share candies, as follows:

- $(0, 1, 3)$
- $(0, 2, 2)$
- $(1, 0, 3)$
- $(1, 1, 2)$
- $(1, 2, 1)$

Here, in each sequence, the i-th element represents the number of candies that Child i receives.

Sample Input 2

```
1 10
9
```

Sample Output 2

```
0
```

There may be no ways for the children to share candies.

Sample Input 3

```
2 0
0 0
```

Sample Output 3

```
1
```

There is one way for the children to share candies, as follows:

- $(0, 0)$

Sample Input 4

```
4 100000
100000 100000 100000 100000
```

Sample Output 4

```
665683269
```

Be sure to print the answer modulo $10^9 + 7$.

N - Slimes

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N slimes lining up in a row. Initially, the i-th slime from the left has a size of a_i.

Taro is trying to combine all the slimes into a larger slime. He will perform the following operation repeatedly until there is only one slime:

- Choose two adjacent slimes, and combine them into a new slime. The new slime has a size of $x + y$, where x and y are the sizes of the slimes before combining them. Here, a cost of $x + y$ is incurred. The positional relationship of the slimes does not change while combining slimes.

Find the minimum possible total cost incurred.

Constraints

- All values in input are integers.
- $2 \leq N \leq 400$
- $1 \leq a_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
a_1 \quad a_2 \quad \ldots \quad a_N
\end{array}
$$

Output

Print the minimum possible total cost incurred.

Sample Input 1

```
4
10 20 30 40
```

Sample Output 1

```
190
```

Taro should do as follows (slimes being combined are shown in bold):

- (**10**, **20**, 30, 40) → (**30**, 30, 40)
- (**30**, **30**, 40) → (**60**, 40)
- (**60**, **40**) → (**100**)

Sample Input 2

```
5
10 10 10 10 10
```

Sample Output 2

```
120
```

Taro should do, for example, as follows:

- (**10**, **10**, 10, 10, 10) → (**20**, 10, 10, 10)
- (20, **10**, **10**, 10) → (20, **20**, 10)
- (20, **20**, **10**) → (20, **30**)
- (**20**, **30**) → (**50**)

Sample Input 3

```
3
1000000000 1000000000 1000000000
```

Sample Output 3

```
5000000000
```

The answer may not fit into a 32-bit integer type.

Sample Input 4

```
6
7 6 8 6 1 1
```

Sample Output 4

```
68
```

Taro should do, for example, as follows:

- $(7, 6, 8, 6, \mathbf{1}, \mathbf{1}) \to (7, 6, 8, 6, \mathbf{2})$
- $(7, 6, 8, \mathbf{6}, \mathbf{2}) \to (7, 6, 8, \mathbf{8})$
- $(\mathbf{7}, \mathbf{6}, 8, 8) \to (\mathbf{13}, 8, 8)$
- $(13, \mathbf{8}, \mathbf{8}) \to (13, \mathbf{16})$
- $(\mathbf{13}, \mathbf{16}) \to (\mathbf{29})$

O - Matching

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N men and N women, both numbered $1, 2, \ldots, N$.

For each i, j ($1 \leq i, j \leq N$), the compatibility of Man i and Woman j is given as an integer $a_{i,j}$. If $a_{i,j} = 1$, Man i and Woman j are compatible; if $a_{i,j} = 0$, they are not.

Taro is trying to make N pairs, each consisting of a man and a woman who are compatible. Here, each man and each woman must belong to exactly one pair.

Find the number of ways in which Taro can make N pairs, modulo $10^9 + 7$.

Constraints

- All values in input are integers.
- $1 \leq N \leq 21$
- $a_{i,j}$ is 0 or 1.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
a_{1,1} \ \ldots \ a_{1,N} \\
: \\
a_{N,1} \ \ldots \ a_{N,N}
\end{array}
$$

Output

Print the number of ways in which Taro can make N pairs, modulo $10^9 + 7$.

Sample Input 1

```
3
0 1 1
1 0 1
1 1 1
```

Sample Output 1

```
3
```

There are three ways to make pairs, as follows ((i, j) denotes a pair of Man i and Woman j):

- $(1, 2), (2, 1), (3, 3)$
- $(1, 2), (2, 3), (3, 1)$
- $(1, 3), (2, 1), (3, 2)$

Sample Input 2

```
4
0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0
```

Sample Output 2

```
1
```

There is one way to make pairs, as follows:

- $(1, 2), (2, 4), (3, 1), (4, 3)$

Sample Input 3

```
1
0
```

Sample Output 3

```
0
```

Sample Input 4

```
21
0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 0 0 1 0 0 1
1 1 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 1 0
0 0 1 1 1 1 0 1 1 0 0 1 0 0 1 1 0 0 0 1 1
0 1 1 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0
1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 1 1 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 0 1
0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 1 0 1 0
0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1
0 0 1 0 0 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1
0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 1
0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 0 1 1 0
0 0 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 1
0 1 1 0 0 1 1 1 1 0 0 0 1 0 1 1 0 1 0 1 1
1 1 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 1
0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1
1 0 1 1 0 1 0 1 0 0 1 0 0 1 1 0 1 0 1 1 0
0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 1
0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 0 0 1 1 0 1
0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 1 0 0 1 1 0
1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 1 0
1 0 0 1 1 0 1 1 1 1 1 0 1 0 1 1 0 0 0 0 0
```

Sample Output 4

```
102515160
```

Be sure to print the number modulo $10^9 + 7$.

P - Independent Set

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a tree with N vertices, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N - 1$), the i-th edge connects Vertex x_i and y_i.

Taro has decided to paint each vertex in white or black. Here, it is not allowed to paint two adjacent vertices both in black.

Find the number of ways in which the vertices can be painted, modulo $10^9 + 7$.

Constraints

- All values in input are integers.
- $1 \leq N \leq 10^5$
- $1 \leq x_i, y_i \leq N$
- The given graph is a tree.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
x_1 \quad y_1 \\
x_2 \quad y_2 \\
: \\
x_{N-1} \quad y_{N-1}
\end{array}
$$

Output

Print the number of ways in which the vertices can be painted, modulo $10^9 + 7$.

Sample Input 1

```
3
1 2
2 3
```

Sample Output 1

```
5
```

There are five ways to paint the vertices, as follows:

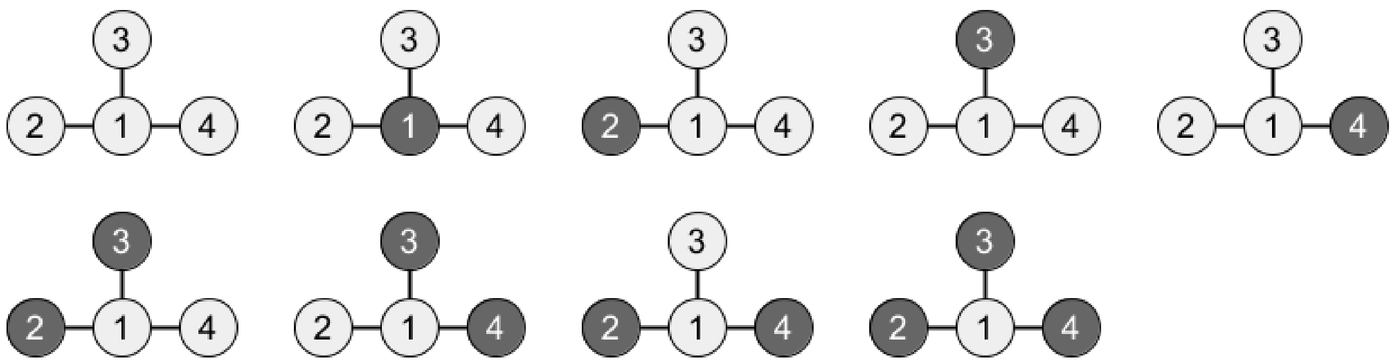
Sample Input 2

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

Sample Output 2

```
9
```

There are nine ways to paint the vertices, as follows:



Sample Input 3

```
1
```

Sample Output 3

```
2
```

Sample Input 4

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

Sample Output 4

```
157
```

Q - Flowers

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N flowers arranged in a row. For each i ($1 \leq i \leq N$), the height and the beauty of the i-th flower from the left is h_i and a_i, respectively. Here, $h_1, h_2, \ldots, h_N$ are all distinct.

Taro is pulling out some flowers so that the following condition is met:

- The heights of the remaining flowers are monotonically increasing from left to right.

Find the maximum possible sum of the beauties of the remaining flowers.

Constraints

- All values in input are integers.
- $1 \leq N \leq 2 \times 10^5$
- $1 \leq h_i \leq N$
- $h_1, h_2, \ldots, h_N$ are all distinct.
- $1 \leq a_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N \\
h_1 & h_2 & \ldots & h_N \\
a_1 & a_2 & \ldots & a_N
\end{array}
$$

Output

Print the maximum possible sum of the beauties of the remaining flowers.

Sample Input 1

```
4
3 1 4 2
10 20 30 40
```

Sample Output 1

```
60
```

We should keep the second and fourth flowers from the left. Then, the heights would be $1, 2$ from left to right, which is monotonically increasing, and the sum of the beauties would be $20 + 40 = 60$.

Sample Input 2

```
1
1
10
```

Sample Output 2

```
10
```

The condition is met already at the beginning.

Sample Input 3

```
5
1 2 3 4 5
1000000000 1000000000 1000000000 1000000000 1000000000
```

Sample Output 3

```
5000000000
```

The answer may not fit into a 32-bit integer type.

Sample Input 4

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

Sample Output 4

```
31
```

We should keep the second, third, sixth, eighth and ninth flowers from the left.

R - Walk

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a simple directed graph G with N vertices, numbered $1, 2, \ldots, N$.

For each i and j ($1 \leq i, j \leq N$), you are given an integer $a_{i,j}$ that represents whether there is a directed edge from Vertex i to j. If $a_{i,j} = 1$, there is a directed edge from Vertex i to j; if $a_{i,j} = 0$, there is not.

Find the number of different directed paths of length K in G, modulo $10^9 + 7$. We will also count a path that traverses the same edge multiple times.

Constraints

- All values in input are integers.
- $1 \leq N \leq 50$
- $1 \leq K \leq 10^{18}$
- $a_{i,j}$ is 0 or 1.
- $a_{i,i} = 0$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad K \\
a_{1,1} \quad \ldots \quad a_{1,N} \\
: \\
a_{N,1} \quad \ldots \quad a_{N,N}
\end{array}
$$

Output

Print the number of different directed paths of length K in G, modulo $10^9 + 7$.

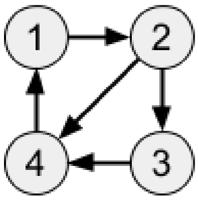
Sample Input 1

```
4 2
0 1 0 0
0 0 1 1
0 0 0 1
1 0 0 0
```

Sample Output 1

```
6
```

G is drawn in the figure below:



There are six directed paths of length 2:

- $1 \rightarrow 2 \rightarrow 3$
- $1 \rightarrow 2 \rightarrow 4$
- $2 \rightarrow 3 \rightarrow 4$
- $2 \rightarrow 4 \rightarrow 1$
- $3 \rightarrow 4 \rightarrow 1$
- $4 \rightarrow 1 \rightarrow 2$

Sample Input 2

```
3 3
0 1 0
1 0 1
0 0 0
```

Sample Output 2

```
3
```

G is drawn in the figure below:

There are three directed paths of length 3:

- $1 \to 2 \to 1 \to 2$
- $2 \to 1 \to 2 \to 1$
- $2 \to 1 \to 2 \to 3$

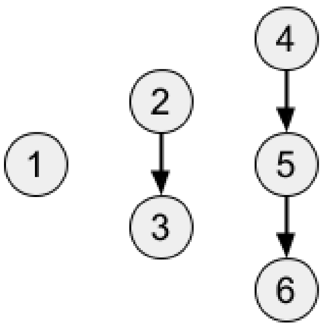
Sample Input 3

```
6 2
0 0 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 0
0 0 0 0 1 0
0 0 0 0 0 1
0 0 0 0 0 0
```

Sample Output 3

```
1
```

G is drawn in the figure below:



There is one directed path of length 2:

- $4 \to 5 \to 6$

Sample Input 4

```
1 1
0
```

Sample Output 4

```
0
```

Sample Input 5

```
10 1000000000000000000
0 0 1 1 0 0 0 1 1 0
0 0 0 0 0 1 1 1 0 0
0 1 0 0 0 1 0 1 0 1
1 1 1 0 1 1 0 1 1 0
0 1 1 1 0 1 0 1 1 1
0 0 0 1 0 0 1 0 1 0
0 0 0 1 1 0 0 1 0 1
1 0 0 0 1 0 1 0 0 0
0 0 0 0 0 1 0 0 0 0
1 0 1 1 1 0 1 1 1 0
```

Sample Output 5

```
957538352
```

Be sure to print the count modulo $10^9 + 7$.

S - Digit Sum

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Find the number of integers between 1 and K (inclusive) satisfying the following condition, modulo $10^9 + 7$:

- The sum of the digits in base ten is a multiple of D.

Constraints

- All values in input are integers.
- $1 \leq K < 10^{10000}$
- $1 \leq D \leq 100$

Input

Input is given from Standard Input in the following format:

```
K
D
```

Output

Print the number of integers satisfying the condition, modulo $10^9 + 7$.

Sample Input 1

```
30
4
```

Sample Output 1

```
6
```

Those six integers are: $4, 8, 13, 17, 22$ and 26.

Sample Input 2

```
1000000009
1
```

Sample Output 2

```
2
```

Be sure to print the number modulo $10^9 + 7$.

Sample Input 3

```
98765432109876543210
58
```

Sample Output 3

```
635270834
```

T - Permutation

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Let N be a positive integer. You are given a string s of length $N-1$, consisting of < and >.

Find the number of permutations $(p_1, p_2, \ldots, p_N)$ of $(1, 2, \ldots, N)$ that satisfy the following condition, modulo $10^9 + 7$:

- For each i ($1 \leq i \leq N-1$), $p_i < p_{i+1}$ if the i-th character in s is <, and $p_i > p_{i+1}$ if the i-th character in s is >.

Constraints

- N is an integer.
- $2 \leq N \leq 3000$
- s is a string of length $N-1$.
- s consists of < and >.

Input

Input is given from Standard Input in the following format:

$$N$$
$$s$$

Output

Print the number of permutations that satisfy the condition, modulo $10^9 + 7$.

Sample Input 1

```
4
<><
```

Sample Output 1

```
5
```

There are five permutations that satisfy the condition, as follows:

- $(1, 3, 2, 4)$
- $(1, 4, 2, 3)$
- $(2, 3, 1, 4)$
- $(2, 4, 1, 3)$
- $(3, 4, 1, 2)$

Sample Input 2

```
5
<<<<
```

Sample Output 2

```
1
```

There is one permutation that satisfies the condition, as follows:

- $(1, 2, 3, 4, 5)$

Sample Input 3

```
20
>>>><>>><>><>>><<>>
```

Sample Output 3

```
217136290
```

Be sure to print the number modulo $10^9 + 7$.

U - Grouping

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N rabbits, numbered $1, 2, \ldots, N$.

For each i, j ($1 \leq i, j \leq N$), the compatibility of Rabbit i and j is described by an integer $a_{i,j}$. Here, $a_{i,i} = 0$ for each i ($1 \leq i \leq N$), and $a_{i,j} = a_{j,i}$ for each i and j ($1 \leq i, j \leq N$).

Taro is dividing the N rabbits into some number of groups. Here, each rabbit must belong to exactly one group. After grouping, for each i and j ($1 \leq i < j \leq N$), Taro earns $a_{i,j}$ points if Rabbit i and j belong to the same group.

Find Taro's maximum possible total score.

Constraints

- All values in input are integers.
- $1 \leq N \leq 16$
- $|a_{i,j}| \leq 10^9$
- $a_{i,i} = 0$
- $a_{i,j} = a_{j,i}$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
a_{1,1} \ \ldots \ a_{1,N} \\
: \\
a_{N,1} \ \ldots \ a_{N,N}
\end{array}
$$

Output

Print Taro's maximum possible total score.

Sample Input 1

```
3
0 10 20
10 0 -100
20 -100 0
```

Sample Output 1

```
20
```

The rabbits should be divided as $\{1, 3\}, \{2\}$.

Sample Input 2

```
2
0 -10
-10 0
```

Sample Output 2

```
0
```

The rabbits should be divided as $\{1\}, \{2\}$.

Sample Input 3

```
4
0 1000000000 1000000000 1000000000
1000000000 0 1000000000 1000000000
1000000000 1000000000 0 -1
1000000000 1000000000 -1 0
```

Sample Output 3

```
4999999999
```

The rabbits should be divided as $\{1, 2, 3, 4\}$. Note that the answer may not fit into a 32-bit integer type.

Sample Input 4

```
16
0 5 -4 -5 -8 -4 7 2 -4 0 7 0 2 -3 7 7
5 0 8 -9 3 5 2 -7 2 -7 0 -1 -4 1 -1 9
-4 8 0 -9 8 9 3 1 4 9 6 6 -6 1 8 9
-5 -9 -9 0 -7 6 4 -1 9 -3 -5 0 1 2 -4 1
-8 3 8 -7 0 -5 -9 9 1 -9 -6 -3 -8 3 4 3
-4 5 9 6 -5 0 -6 1 -2 2 0 -5 -2 3 1 2
7 2 3 4 -9 -6 0 -2 -2 -9 -3 9 -2 9 2 -5
2 -7 1 -1 9 1 -2 0 -6 0 -6 6 4 -1 -7 8
-4 2 4 9 1 -2 -2 -6 0 8 -6 -2 -4 8 7 7
0 -7 9 -3 -9 2 -9 0 8 0 0 1 -3 3 -6 -6
7 0 6 -5 -6 0 -3 -6 -6 0 0 5 7 -1 -5 3
0 -1 6 0 -3 -5 9 6 -2 1 5 0 -2 7 -8 0
2 -4 -6 1 -8 -2 -2 4 -4 -3 7 -2 0 -9 7 1
-3 1 1 2 3 3 9 -1 8 3 -1 7 -9 0 -6 -8
7 -1 8 -4 4 1 2 -7 7 -6 -5 -8 7 -6 0 -9
7 9 9 1 3 2 -5 8 7 -6 3 0 1 -8 -9 0
```

Sample Output 4

```
132
```

V - Subtree

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a tree with N vertices, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N - 1$), the i-th edge connects Vertex x_i and y_i.

Taro has decided to paint each vertex in white or black, so that any black vertex can be reached from any other black vertex by passing through only black vertices.

You are given a positive integer M. For each v ($1 \leq v \leq N$), answer the following question:

- Assuming that Vertex v has to be black, find the number of ways in which the vertices can be painted, modulo M.

Constraints

- All values in input are integers.
- $1 \leq N \leq 10^5$
- $2 \leq M \leq 10^9$
- $1 \leq x_i, y_i \leq N$
- The given graph is a tree.

Input

Input is given from Standard Input in the following format:

```
N M
x_1 y_1
x_2 y_2
:
x_{N-1} y_{N-1}
```

Output

Print N lines. The v-th ($1 \leq v \leq N$) line should contain the answer to the following question:

- Assuming that Vertex v has to be black, find the number of ways in which the vertices can be painted, modulo M.

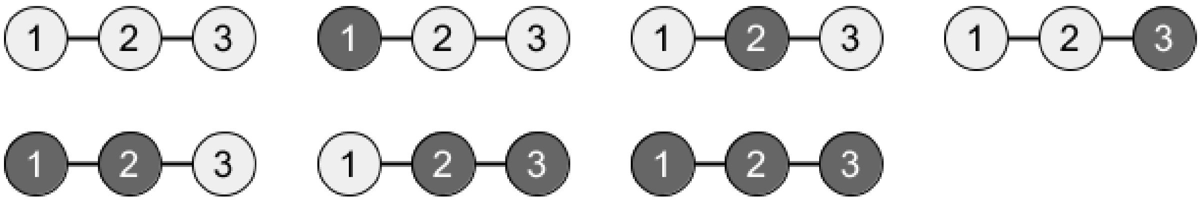
Sample Input 1

```
3 100
1 2
2 3
```

Sample Output 1

```
3
4
3
```

There are seven ways to paint the vertices, as shown in the figure below. Among them, there are three ways such that Vertex 1 is black, four ways such that Vertex 2 is black and three ways such that Vertex 3 is black.



Sample Input 2

```
4 100
1 2
1 3
1 4
```

Sample Output 2

```
8
5
5
5
```

Sample Input 3

```
1 100
```

Sample Output 3

```
1
```

Sample Input 4

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

Sample Output 4

```
0
0
1
1
1
0
1
0
1
1
```

Be sure to print the answers modulo M.

W - Intervals

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Consider a string of length N consisting of `0` and `1`. The score for the string is calculated as follows:

- For each i ($1 \leq i \leq M$), a_i is added to the score if the string contains `1` at least once between the l_i -th and r_i-th characters (inclusive).

Find the maximum possible score of a string.

Constraints

- All values in input are integers.
- $1 \leq N \leq 2 \times 10^5$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq l_i \leq r_i \leq N$
- $|a_i| \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
l_1 \quad r_1 \quad a_1 \\
l_2 \quad r_2 \quad a_2 \\
: \\
l_M \quad r_M \quad a_M
\end{array}
$$

Output

Print the maximum possible score of a string.

Sample Input 1

```
5 3
1 3 10
2 4 -10
3 5 10
```

Sample Output 1

```
20
```

The score for `10001` is $a_1 + a_3 = 10 + 10 = 20$.

Sample Input 2

```
3 4
1 3 100
1 1 -10
2 2 -20
3 3 -30
```

Sample Output 2

```
90
```

The score for `100` is $a_1 + a_2 = 100 + (-10) = 90$.

Sample Input 3

```
1 1
1 1 -10
```

Sample Output 3

```
0
```

The score for `0` is 0.

Sample Input 4

```
1 5
1 1 1000000000
1 1 1000000000
1 1 1000000000
1 1 1000000000
1 1 1000000000
```

Sample Output 4

```
5000000000
```

The answer may not fit into a 32-bit integer type.

Sample Input 5

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

Sample Output 5

```
10
```

For example, the score for `101000` is $a_2 + a_3 + a_4 + a_5 + a_7 = 10 + (-8) + 5 + 9 + (-6) = 10.$

X - Tower

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N blocks, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), Block i has a weight of w_i, a solidness of s_i and a value of v_i.

Taro has decided to build a tower by choosing some of the N blocks and stacking them vertically in some order. Here, the tower must satisfy the following condition:

- For each Block i contained in the tower, the sum of the weights of the blocks stacked above it is not greater than s_i.

Find the maximum possible sum of the values of the blocks contained in the tower.

Constraints

- All values in input are integers.
- $1 \leq N \leq 10^3$
- $1 \leq w_i, s_i \leq 10^4$
- $1 \leq v_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{lll}
N & & \\
w_1 & s_1 & v_1 \\
w_2 & s_2 & v_2 \\
\vdots & & \\
w_N & s_N & v_N
\end{array}
$$

Output

Print the maximum possible sum of the values of the blocks contained in the tower.

Sample Input 1

```
3
2 2 20
2 1 30
3 1 40
```

Sample Output 1

```
50
```

If Blocks $2, 1$ are stacked in this order from top to bottom, this tower will satisfy the condition, with the total value of $30 + 20 = 50$.

Sample Input 2

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

Sample Output 2

```
40
```

Blocks $1, 2, 3, 4$ should be stacked in this order from top to bottom.

Sample Input 3

```
5
1 10000 1000000000
1 10000 1000000000
1 10000 1000000000
1 10000 1000000000
1 10000 1000000000
```

Sample Output 3

```
5000000000
```

The answer may not fit into a 32-bit integer type.

Sample Input 4

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

Sample Output 4

```
22
```

We should, for example, stack Blocks $5, 6, 8, 4$ in this order from top to bottom.

Y - Grid 2

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a grid with H horizontal rows and W vertical columns. Let (i, j) denote the square at the i-th row from the top and the j-th column from the left.

In the grid, N Squares $(r_1, c_1), (r_2, c_2), \ldots, (r_N, c_N)$ are wall squares, and the others are all empty squares. It is guaranteed that Squares $(1, 1)$ and (H, W) are empty squares.

Taro will start from Square $(1, 1)$ and reach (H, W) by repeatedly moving right or down to an adjacent empty square.

Find the number of Taro's paths from Square $(1, 1)$ to (H, W), modulo $10^9 + 7$.

Constraints

- All values in input are integers.
- $2 \leq H, W \leq 10^5$
- $1 \leq N \leq 3000$
- $1 \leq r_i \leq H$
- $1 \leq c_i \leq W$
- Squares (r_i, c_i) are all distinct.
- Squares $(1, 1)$ and (H, W) are empty squares.

Input

Input is given from Standard Input in the following format:

```
H W N
r_1 c_1
r_2 c_2
:
r_N c_N
```

Output

Print the number of Taro's paths from Square $(1, 1)$ to (H, W), modulo $10^9 + 7$.

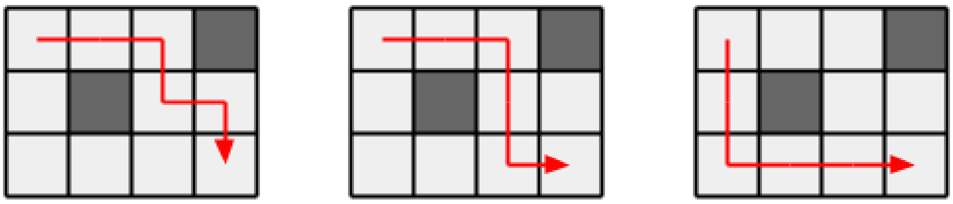
Sample Input 1

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

Sample Output 1

```
3
```

There are three paths as follows:



Sample Input 2

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

Sample Output 2

```
0
```

There may be no paths.

Sample Input 3

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

Sample Output 3

```
24
```

Sample Input 4

```
100000 100000 1
50000 50000
```

Sample Output 4

```
123445622
```

Be sure to print the count modulo $10^9 + 7$.

Z - Frog 3

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N stones, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), the height of Stone i is h_i. Here, $h_1 < h_2 < \cdots < h_N$ holds.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N:

- If the frog is currently on Stone i, jump to one of the following: Stone $i + 1, i + 2, \ldots, N$. Here, a cost of $(h_j - h_i)^2 + C$ is incurred, where j is the stone to land on.

Find the minimum possible total cost incurred before the frog reaches Stone N.

Constraints

- All values in input are integers.
- $2 \leq N \leq 2 \times 10^5$
- $1 \leq C \leq 10^{12}$
- $1 \leq h_1 < h_2 < \cdots < h_N \leq 10^6$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad C \\
h_1 \quad h_2 \quad \ldots \quad h_N
\end{array}
$$

Output

Print the minimum possible total cost incurred.

Sample Input 1

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

Sample Output 1

```
20
```

If we follow the path $1 \to 3 \to 5$, the total cost incurred would be $((3-1)^2+6)+((5-3)^2+6) = 20$.

Sample Input 2

```
2 1000000000000
500000 1000000
```

Sample Output 2

```
1250000000000
```

The answer may not fit into a 32-bit integer type.

Sample Input 3

```
8 5
1 3 4 5 10 11 12 13
```

Sample Output 3

```
62
```

If we follow the path $1 \to 2 \to 4 \to 5 \to 8$, the total cost incurred would be $((3-1)^2+5)+((5-3)^2+5)+((10-5)^2+5)+((13-10)^2+5) = 62$.