**A - Frog 1**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN stones, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), the height of Stone ii is hihi.

There is a frog who is initially on Stone 11. He will repeat the following action some number of times to reach Stone NN:

* If the frog is currently on Stone ii, jump to Stone i+1i+1 or Stone i+2i+2. Here, a cost of |hi−hj||hi−hj| is incurred, where jj is the stone to land on.

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

**Constraints**

* All values in input are integers.
* 2≤N≤1052≤N≤105
* 1≤hi≤1041≤hi≤104

**Input**

Input is given from Standard Input in the following format:

NN

h1h1 h2h2 …… hNhN

**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 11 → 22 → 44, the total cost incurred would be |10−30|+|30−20|=30|10−30|+|30−20|=30.

**Sample Input 2**

2

10 10

**Sample Output 2**

0

If we follow the path 11 → 22, the total cost incurred would be |10−10|=0|10−10|=0.

**Sample Input 3**

6

30 10 60 10 60 50

**Sample Output 3**

40

If we follow the path 11 → 33 → 55 → 66, the total cost incurred would be |30−60|+|60−60|+|60−50|=40|30−60|+|60−60|+|60−50|=40.

**B - Frog 2**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN stones, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), the height of Stone ii is hihi.

There is a frog who is initially on Stone 11. He will repeat the following action some number of times to reach Stone NN:

* If the frog is currently on Stone ii, jump to one of the following: Stone i+1,i+2,…,i+Ki+1,i+2,…,i+K. Here, a cost of |hi−hj||hi−hj| is incurred, where jj is the stone to land on.

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

**Constraints**

* All values in input are integers.
* 2≤N≤1052≤N≤105
* 1≤K≤1001≤K≤100
* 1≤hi≤1041≤hi≤104

**Input**

Input is given from Standard Input in the following format:

NN KK

h1h1 h2h2 …… hNhN

**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 11 → 22 → 55, the total cost incurred would be |10−30|+|30−20|=30|10−30|+|30−20|=30.

**Sample Input 2**

3 1

10 20 10

**Sample Output 2**

20

If we follow the path 11 → 22 → 33, the total cost incurred would be |10−20|+|20−10|=20|10−20|+|20−10|=20.

**Sample Input 3**

2 100

10 10

**Sample Output 3**

0

If we follow the path 11 → 22, the total cost incurred would be |10−10|=0|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 11 → 44 → 88 → 1010, the total cost incurred would be |40−70|+|70−70|+|70−60|=40|40−70|+|70−70|+|70−60|=40.

**C - Vacation**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

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

The vacation consists of NN days. For each ii (1≤i≤N1≤i≤N), Taro will choose one of the following activities and do it on the ii-th day:

* A: Swim in the sea. Gain aiai points of happiness.
* B: Catch bugs in the mountains. Gain bibi points of happiness.
* C: Do homework at home. Gain cici 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≤N≤1051≤N≤105
* 1≤ai,bi,ci≤1041≤ai,bi,ci≤104

**Input**

Input is given from Standard Input in the following format:

NN

a1a1 b1b1 c1c1

a2a2 b2b2 c2c2

::

aNaN bNbN cNcN

**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=21070+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 : 100100 points

**Problem Statement**

There are NN items, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), Item ii has a weight of wiwi and a value of vivi.

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

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

**Constraints**

* All values in input are integers.
* 1≤N≤1001≤N≤100
* 1≤W≤1051≤W≤105
* 1≤wi≤W1≤wi≤W
* 1≤vi≤1091≤vi≤109

**Input**

Input is given from Standard Input in the following format:

NN WW

w1w1 v1v1

w2w2 v2v2

::

wNwN vNvN

**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 11 and 33 should be taken. Then, the sum of the weights is 3+5=83+5=8, and the sum of the values is 30+60=9030+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 2,42,4 and 55 should be taken. Then, the sum of the weights is 5+6+3=145+6+3=14, and the sum of the values is 6+6+5=176+6+5=17.

**E - Knapsack 2**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN items, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), Item ii has a weight of wiwi and a value of vivi.

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

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

**Constraints**

* All values in input are integers.
* 1≤N≤1001≤N≤100
* 1≤W≤1091≤W≤109
* 1≤wi≤W1≤wi≤W
* 1≤vi≤1031≤vi≤103

**Input**

Input is given from Standard Input in the following format:

NN WW

w1w1 v1v1

w2w2 v2v2

::

wNwN vNvN

**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 11 and 33 should be taken. Then, the sum of the weights is 3+5=83+5=8, and the sum of the values is 30+60=9030+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 2,42,4 and 55 should be taken. Then, the sum of the weights is 5+6+3=145+6+3=14, and the sum of the values is 6+6+5=176+6+5=17.

**F - LCS**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

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

**Notes**

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

**Constraints**

* ss and tt are strings consisting of lowercase English letters.
* 1≤|s|,|t|≤30001≤|s|,|t|≤3000

**Input**

Input is given from Standard Input in the following format:

ss

tt

**Output**

Print one longest string that is a subsequence of both ss and tt. 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 : 100100 points

**Problem Statement**

There is a directed graph GG with NN vertices and MM edges. The vertices are numbered 1,2,…,N1,2,…,N, and for each ii (1≤i≤M1≤i≤M), the ii-th directed edge goes from Vertex xixi to yiyi. GG **does not contain directed cycles**.

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

**Constraints**

* All values in input are integers.
* 2≤N≤1052≤N≤105
* 1≤M≤1051≤M≤105
* 1≤xi,yi≤N1≤xi,yi≤N
* All pairs (xi,yi)(xi,yi) are distinct.
* GG **does not contain directed cycles**.

**Input**

Input is given from Standard Input in the following format:

NN MM

x1x1 y1y1

x2x2 y2y2

::

xMxM yMyM

**Output**

Print the length of the longest directed path in GG.

**Sample Input 1**

4 5

1 2

1 3

3 2

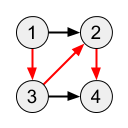
2 4

3 4

**Sample Output 1**

3

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



**Sample Input 2**

6 3

2 3

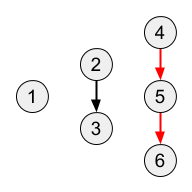
4 5

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

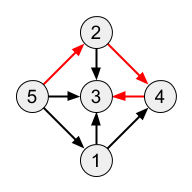
4 3

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 : 100100 points

**Problem Statement**

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

For each ii and jj (1≤i≤H1≤i≤H, 1≤j≤W1≤j≤W), Square (i,j)(i,j) is described by a character ai,jai,j. If ai,jai,j is ., Square (i,j)(i,j) is an empty square; if ai,jai,j is #, Square (i,j)(i,j) is a wall square. It is guaranteed that Squares (1,1)(1,1) and (H,W)(H,W) are empty squares.

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

Find the number of Taro's paths from Square (1,1)(1,1) to (H,W)(H,W). As the answer can be extremely large, find the count modulo 109+7109+7.

**Constraints**

* HH and WW are integers.
* 2≤H,W≤10002≤H,W≤1000
* ai,jai,j is . or #.
* Squares (1,1)(1,1) and (H,W)(H,W) are empty squares.

**Input**

Input is given from Standard Input in the following format:

HH WW

a1,1a1,1……a1,Wa1,W

::

aH,1aH,1……aH,WaH,W

**Output**

Print the number of Taro's paths from Square (1,1)(1,1) to (H,W)(H,W), modulo 109+7109+7.

**Sample Input 1**

3 4

...#

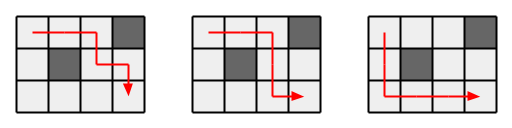
.#..

....

**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 109+7109+7.

**I - Coins**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

Let NN be a positive odd number.

There are NN coins, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), when Coin ii is tossed, it comes up heads with probability pipi and tails with probability 1−pi1−pi.

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

**Constraints**

* NN is an odd number.
* 1≤N≤29991≤N≤2999
* pipi is a real number and has two decimal places.
* 0<pi<10<pi<1

**Input**

Input is given from Standard Input in the following format:

NN

p1p1 p2p2 …… pNpN

**Output**

Print the probability of having more heads than tails. The output is considered correct when the absolute error is not greater than 10−910−9.

**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)(Coin1,Coin2,Coin3)=(Head,Head,Head) is 0.3×0.6×0.8=0.1440.3×0.6×0.8=0.144;
* The probability of having (Coin1,Coin2,Coin3)=(Tail,Head,Head)(Coin1,Coin2,Coin3)=(Tail,Head,Head) is 0.7×0.6×0.8=0.3360.7×0.6×0.8=0.336;
* The probability of having (Coin1,Coin2,Coin3)=(Head,Tail,Head)(Coin1,Coin2,Coin3)=(Head,Tail,Head) is 0.3×0.4×0.8=0.0960.3×0.4×0.8=0.096;
* The probability of having (Coin1,Coin2,Coin3)=(Head,Head,Tail)(Coin1,Coin2,Coin3)=(Head,Head,Tail) is 0.3×0.6×0.2=0.0360.3×0.6×0.2=0.036.

Thus, the probability of having more heads than tails is 0.144+0.336+0.096+0.036=0.6120.144+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 : 100100 points

**Problem Statement**

There are NN dishes, numbered 1,2,…,N1,2,…,N. Initially, for each ii (1≤i≤N1≤i≤N), Dish ii has aiai (1≤ai≤31≤ai≤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,…,N1,2,…,N with equal probabilities, and let ii be the outcome. If there are some pieces of sushi on Dish ii, 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≤N≤3001≤N≤300
* 1≤ai≤31≤ai≤3

**Input**

Input is given from Standard Input in the following format:

NN

a1a1 a2a2 …… aNaN

**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 10−910−9.

**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 11. After that, the expected number of operations before the second sushi is eaten, is 1.51.5. After that, the expected number of operations before the third sushi is eaten, is 33. Thus, the expected total number of operations is 1+1.5+3=5.51+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 : 100100 points

**Problem Statement**

There is a set A={a1,a2,…,aN}A={a1,a2,…,aN} consisting of NN positive integers. Taro and Jiro will play the following game against each other.

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

* Choose an element xx in AA, and remove exactly xx 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≤N≤1001≤N≤100
* 1≤K≤1051≤K≤105
* 1≤a1<a2<⋯<aN≤K1≤a1<a2<⋯<aN≤K

**Input**

Input is given from Standard Input in the following format:

NN KK

a1a1 a2a2 …… aNaN

**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 : 100100 points

**Problem Statement**

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

Initially, they are given a sequence a=(a1,a2,…,aN)a=(a1,a2,…,aN). Until aa becomes empty, the two players perform the following operation alternately, starting from Taro:

* Remove the element at the beginning or the end of aa. The player earns xx points, where xx is the removed element.

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

Assuming that the two players play optimally, find the resulting value of X−YX−Y.

**Constraints**

* All values in input are integers.
* 1≤N≤30001≤N≤3000
* 1≤ai≤1091≤ai≤109

**Input**

Input is given from Standard Input in the following format:

NN

a1a1 a2a2 …… aNaN

**Output**

Print the resulting value of X−YX−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**) → (10, 80, 90)
* Jiro: (10, 80, **90**) → (10, 80)
* Taro: (10, **80**) → (10)
* Jiro: (**10**) → ()

Here, X=30+80=110X=30+80=110 and Y=90+10=100Y=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) → (100, 10)
* Jiro: (**100**, 10) → (10)
* Taro: (**10**) → ()

Here, X=10+10=20X=10+10=20 and Y=100Y=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=15X=5+1+9=15 and Y=4+7+2=13Y=4+7+2=13.

**M - Candies**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN children, numbered 1,2,…,N1,2,…,N.

They have decided to share KK candies among themselves. Here, for each ii (1≤i≤N1≤i≤N), Child ii must receive between 00 and aiai candies (inclusive). Also, no candies should be left over.

Find the number of ways for them to share candies, modulo 109+7109+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≤N≤1001≤N≤100
* 0≤K≤1050≤K≤105
* 0≤ai≤K0≤ai≤K

**Input**

Input is given from Standard Input in the following format:

NN KK

a1a1 a2a2 …… aNaN

**Output**

Print the number of ways for the children to share candies, modulo 109+7109+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)(0,1,3)
* (0,2,2)(0,2,2)
* (1,0,3)(1,0,3)
* (1,1,2)(1,1,2)
* (1,2,1)(1,2,1)

Here, in each sequence, the ii-th element represents the number of candies that Child ii 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)(0,0)

**Sample Input 4**

4 100000

100000 100000 100000 100000

**Sample Output 4**

665683269

Be sure to print the answer modulo 109+7109+7.

**N - Slimes**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN slimes lining up in a row. Initially, the ii-th slime from the left has a size of aiai.

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+yx+y, where xx and yy are the sizes of the slimes before combining them. Here, a cost of x+yx+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≤N≤4002≤N≤400
* 1≤ai≤1091≤ai≤109

**Input**

Input is given from Standard Input in the following format:

NN

a1a1 a2a2 …… aNaN

**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, **1**, **1**) → (7, 6, 8, 6, **2**)
* (7, 6, 8, **6**, **2**) → (7, 6, 8, **8**)
* (**7**, **6**, 8, 8) → (**13**, 8, 8)
* (13, **8**, **8**) → (13, **16**)
* (**13**, **16**) → (**29**)

**O - Matching**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN men and NN women, both numbered 1,2,…,N1,2,…,N.

For each i,ji,j (1≤i,j≤N1≤i,j≤N), the compatibility of Man ii and Woman jj is given as an integer ai,jai,j. If ai,j=1ai,j=1, Man ii and Woman jj are compatible; if ai,j=0ai,j=0, they are not.

Taro is trying to make NN 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 NN pairs, modulo 109+7109+7.

**Constraints**

* All values in input are integers.
* 1≤N≤211≤N≤21
* ai,jai,j is 00 or 11.

**Input**

Input is given from Standard Input in the following format:

NN

a1,1a1,1 …… a1,Na1,N

::

aN,1aN,1 …… aN,NaN,N

**Output**

Print the number of ways in which Taro can make NN pairs, modulo 109+7109+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)(i,j) denotes a pair of Man ii and Woman jj):

* (1,2),(2,1),(3,3)(1,2),(2,1),(3,3)
* (1,2),(2,3),(3,1)(1,2),(2,3),(3,1)
* (1,3),(2,1),(3,2)(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)(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 109+7109+7.

**P - Independent Set**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There is a tree with NN vertices, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N−11≤i≤N−1), the ii-th edge connects Vertex xixi and yiyi.

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 109+7109+7.

**Constraints**

* All values in input are integers.
* 1≤N≤1051≤N≤105
* 1≤xi,yi≤N1≤xi,yi≤N
* The given graph is a tree.

**Input**

Input is given from Standard Input in the following format:

NN

x1x1 y1y1

x2x2 y2y2

::

xN−1xN−1 yN−1yN−1

**Output**

Print the number of ways in which the vertices can be painted, modulo 109+7109+7.

**Sample Input 1**

3

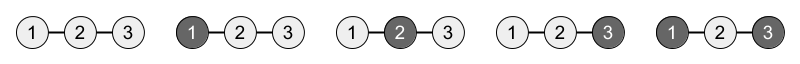
1 2

2 3

**Sample Output 1**

5

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



**Sample Input 2**

4

1 2

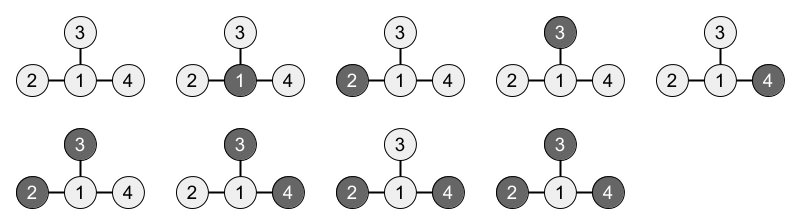
1 3

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 : 100100 points

**Problem Statement**

There are NN flowers arranged in a row. For each ii (1≤i≤N1≤i≤N), the height and the beauty of the ii-th flower from the left is hihi and aiai, respectively. Here, h1,h2,…,hNh1,h2,…,hN 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≤N≤2×1051≤N≤2×105
* 1≤hi≤N1≤hi≤N
* h1,h2,…,hNh1,h2,…,hN are all distinct.
* 1≤ai≤1091≤ai≤109

**Input**

Input is given from Standard Input in the following format:

NN

h1h1 h2h2 …… hNhN

a1a1 a2a2 …… aNaN

**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 1,21,2 from left to right, which is monotonically increasing, and the sum of the beauties would be 20+40=6020+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 : 100100 points

**Problem Statement**

There is a simple directed graph GG with NN vertices, numbered 1,2,…,N1,2,…,N.

For each ii and jj (1≤i,j≤N1≤i,j≤N), you are given an integer ai,jai,j that represents whether there is a directed edge from Vertex ii to jj. If ai,j=1ai,j=1, there is a directed edge from Vertex ii to jj; if ai,j=0ai,j=0, there is not.

Find the number of different directed paths of length KK in GG, modulo 109+7109+7. We will also count a path that traverses the same edge multiple times.

**Constraints**

* All values in input are integers.
* 1≤N≤501≤N≤50
* 1≤K≤10181≤K≤1018
* ai,jai,j is 00 or 11.
* ai,i=0ai,i=0

**Input**

Input is given from Standard Input in the following format:

NN KK

a1,1a1,1 …… a1,Na1,N

::

aN,1aN,1 …… aN,NaN,N

**Output**

Print the number of different directed paths of length KK in GG, modulo 109+7109+7.

**Sample Input 1**

4 2

0 1 0 0

0 0 1 1

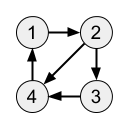
0 0 0 1

1 0 0 0

**Sample Output 1**

6

GG is drawn in the figure below:



There are six directed paths of length 22:

* 11 → 22 → 33
* 11 → 22 → 44
* 22 → 33 → 44
* 22 → 44 → 11
* 33 → 44 → 11
* 44 → 11 → 22

**Sample Input 2**

3 3

0 1 0

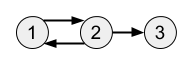
1 0 1

0 0 0

**Sample Output 2**

3

GG is drawn in the figure below:



There are three directed paths of length 33:

* 11 → 22 → 11 → 22
* 22 → 11 → 22 → 11
* 22 → 11 → 22 → 33

**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

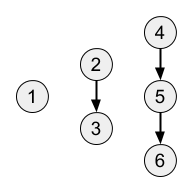
0 0 0 0 0 1

0 0 0 0 0 0

**Sample Output 3**

1

GG is drawn in the figure below:



There is one directed path of length 22:

* 44 → 55 → 66

**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 109+7109+7.

**S - Digit Sum**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

Find the number of integers between 11 and KK (inclusive) satisfying the following condition, modulo 109+7109+7:

* The sum of the digits in base ten is a multiple of DD.

**Constraints**

* All values in input are integers.
* 1≤K<10100001≤K<1010000
* 1≤D≤1001≤D≤100

**Input**

Input is given from Standard Input in the following format:

KK

DD

**Output**

Print the number of integers satisfying the condition, modulo 109+7109+7.

**Sample Input 1**

30

4

**Sample Output 1**

6

Those six integers are: 4,8,13,17,224,8,13,17,22 and 2626.

**Sample Input 2**

1000000009

1

**Sample Output 2**

2

Be sure to print the number modulo 109+7109+7.

**Sample Input 3**

98765432109876543210

58

**Sample Output 3**

635270834

**T - Permutation**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

Let NN be a positive integer. You are given a string ss of length N−1N−1, consisting of < and >.

Find the number of permutations (p1,p2,…,pN)(p1,p2,…,pN) of (1,2,…,N)(1,2,…,N) that satisfy the following condition, modulo 109+7109+7:

* For each ii (1≤i≤N−11≤i≤N−1), pi<pi+1pi<pi+1 if the ii-th character in ss is <, and pi>pi+1pi>pi+1 if the ii-th character in ss is >.

**Constraints**

* NN is an integer.
* 2≤N≤30002≤N≤3000
* ss is a string of length N−1N−1.
* ss consists of < and >.

**Input**

Input is given from Standard Input in the following format:

NN

ss

**Output**

Print the number of permutations that satisfy the condition, modulo 109+7109+7.

**Sample Input 1**

4

<><

**Sample Output 1**

5

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

* (1,3,2,4)(1,3,2,4)
* (1,4,2,3)(1,4,2,3)
* (2,3,1,4)(2,3,1,4)
* (2,4,1,3)(2,4,1,3)
* (3,4,1,2)(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)(1,2,3,4,5)

**Sample Input 3**

20

>>>><>>><>><>>><<>>

**Sample Output 3**

217136290

Be sure to print the number modulo 109+7109+7.

**U - Grouping**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN rabbits, numbered 1,2,…,N1,2,…,N.

For each i,ji,j (1≤i,j≤N1≤i,j≤N), the compatibility of Rabbit ii and jj is described by an integer ai,jai,j. Here, ai,i=0ai,i=0 for each ii (1≤i≤N1≤i≤N), and ai,j=aj,iai,j=aj,i for each ii and jj (1≤i,j≤N1≤i,j≤N).

Taro is dividing the NN rabbits into some number of groups. Here, each rabbit must belong to exactly one group. After grouping, for each ii and jj (1≤i<j≤N1≤i<j≤N), Taro earns ai,jai,j points if Rabbit ii and jj belong to the same group.

Find Taro's maximum possible total score.

**Constraints**

* All values in input are integers.
* 1≤N≤161≤N≤16
* |ai,j|≤109|ai,j|≤109
* ai,i=0ai,i=0
* ai,j=aj,iai,j=aj,i

**Input**

Input is given from Standard Input in the following format:

NN

a1,1a1,1 …… a1,Na1,N

::

aN,1aN,1 …… aN,NaN,N

**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}{1,3},{2}.

**Sample Input 2**

2

0 -10

-10 0

**Sample Output 2**

0

The rabbits should be divided as {1},{2}{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}{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 : 100100 points

**Problem Statement**

There is a tree with NN vertices, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N−11≤i≤N−1), the ii-th edge connects Vertex xixi and yiyi.

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 MM. For each vv (1≤v≤N1≤v≤N), answer the following question:

* Assuming that Vertex vv has to be black, find the number of ways in which the vertices can be painted, modulo MM.

**Constraints**

* All values in input are integers.
* 1≤N≤1051≤N≤105
* 2≤M≤1092≤M≤109
* 1≤xi,yi≤N1≤xi,yi≤N
* The given graph is a tree.

**Input**

Input is given from Standard Input in the following format:

NN MM

x1x1 y1y1

x2x2 y2y2

::

xN−1xN−1 yN−1yN−1

**Output**

Print NN lines. The vv-th (1≤v≤N1≤v≤N) line should contain the answer to the following question:

* Assuming that Vertex vv has to be black, find the number of ways in which the vertices can be painted, modulo MM.

**Sample Input 1**

3 100

1 2

2 3

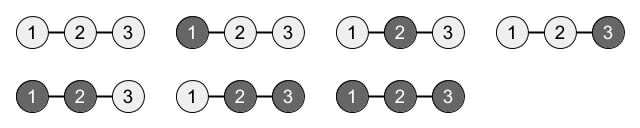
**Sample Output 1**

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 11 is black, four ways such that Vertex 22 is black and three ways such that Vertex 33 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 MM.

**W - Intervals**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

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

* For each ii (1≤i≤M1≤i≤M), aiai is added to the score if the string contains 1 at least once between the lili-th and riri-th characters (inclusive).

Find the maximum possible score of a string.

**Constraints**

* All values in input are integers.
* 1≤N≤2×1051≤N≤2×105
* 1≤M≤2×1051≤M≤2×105
* 1≤li≤ri≤N1≤li≤ri≤N
* |ai|≤109|ai|≤109

**Input**

Input is given from Standard Input in the following format:

NN MM

l1l1 r1r1 a1a1

l2l2 r2r2 a2a2

::

lMlM rMrM aMaM

**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 a1+a3=10+10=20a1+a3=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 a1+a2=100+(−10)=90a1+a2=100+(−10)=90.

**Sample Input 3**

1 1

1 1 -10

**Sample Output 3**

0

The score for 0 is 00.

**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 a2+a3+a4+a5+a7=10+(−8)+5+9+(−6)=10a2+a3+a4+a5+a7=10+(−8)+5+9+(−6)=10.

**X - Tower**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN blocks, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), Block ii has a weight of wiwi, a solidness of sisi and a value of vivi.

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

* For each Block ii contained in the tower, the sum of the weights of the blocks stacked above it is not greater than sisi.

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

**Constraints**

* All values in input are integers.
* 1≤N≤1031≤N≤103
* 1≤wi,si≤1041≤wi,si≤104
* 1≤vi≤1091≤vi≤109

**Input**

Input is given from Standard Input in the following format:

NN

w1w1 s1s1 v1v1

w2w2 s2s2 v2v2

::

wNwN sNsN vNvN

**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 2,12,1 are stacked in this order from top to bottom, this tower will satisfy the condition, with the total value of 30+20=5030+20=50.

**Sample Input 2**

4

1 2 10

3 1 10

2 4 10

1 6 10

**Sample Output 2**

40

Blocks 1,2,3,41,2,3,4 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 5,6,8,45,6,8,4 in this order from top to bottom.

**Y - Grid 2**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

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

In the grid, NN Squares (r1,c1),(r2,c2),…,(rN,cN)(r1,c1),(r2,c2),…,(rN,cN) are wall squares, and the others are all empty squares. It is guaranteed that Squares (1,1)(1,1) and (H,W)(H,W) are empty squares.

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

Find the number of Taro's paths from Square (1,1)(1,1) to (H,W)(H,W), modulo 109+7109+7.

**Constraints**

* All values in input are integers.
* 2≤H,W≤1052≤H,W≤105
* 1≤N≤30001≤N≤3000
* 1≤ri≤H1≤ri≤H
* 1≤ci≤W1≤ci≤W
* Squares (ri,ci)(ri,ci) are all distinct.
* Squares (1,1)(1,1) and (H,W)(H,W) are empty squares.

**Input**

Input is given from Standard Input in the following format:

HH WW NN

r1r1 c1c1

r2r2 c2c2

::

rNrN cNcN

**Output**

Print the number of Taro's paths from Square (1,1)(1,1) to (H,W)(H,W), modulo 109+7109+7.

**Sample Input 1**

3 4 2

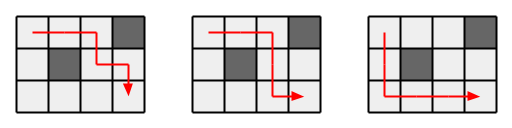
2 2

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 109+7109+7.

**Z - Frog 3**

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100100 points

**Problem Statement**

There are NN stones, numbered 1,2,…,N1,2,…,N. For each ii (1≤i≤N1≤i≤N), the height of Stone ii is hihi. Here, h1<h2<⋯<hNh1<h2<⋯<hN holds.

There is a frog who is initially on Stone 11. He will repeat the following action some number of times to reach Stone NN:

* If the frog is currently on Stone ii, jump to one of the following: Stone i+1,i+2,…,Ni+1,i+2,…,N. Here, a cost of (hj−hi)2+C(hj−hi)2+C is incurred, where jj is the stone to land on.

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

**Constraints**

* All values in input are integers.
* 2≤N≤2×1052≤N≤2×105
* 1≤C≤10121≤C≤1012
* 1≤h1<h2<⋯<hN≤1061≤h1<h2<⋯<hN≤106

**Input**

Input is given from Standard Input in the following format:

NN CC

h1h1 h2h2 …… hNhN

**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 11 → 33 → 55, the total cost incurred would be ((3−1)2+6)+((5−3)2+6)=20((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 11 → 22 → 44 → 55 → 88, the total cost incurred would be ((3−1)2+5)+((5−3)2+5)+((10−5)2+5)+((13−10)2+5)=62((3−1)2+5)+((5−3)2+5)+((10−5)2+5)+((13−10)2+5)=62