Problem B. Best Meeting Places

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

A tree with N vertices is given. Vertices are numbered sequentially from 1 to N. The i-th edge connects vertices A_i and B_i, and has weight C_i, for $1 \le i \le N-1$.

The *teleport distance* between two vertices of the tree is the maximum weight of the edge on the shortest path connecting them. The teleport distance between a vertex and itself is defined as 0.

People living on the tree want to hold N meetings. The i-th meeting is attended by people living in the vertices numbered from 1 to i. This year, because of the spread of coronavirus, the meeting participants will arrive at X selected locations, and then connect via Internet from these locations.

More formally, for each meeting, we will choose X pairwise distinct vertices v_1, v_2, …, v_X. Once the vertices are determined, each person will move to one of the vertices v_1, …, v_X with the minimum teleport distance to it. Let us define the *meeting cost* for the given X and i as the maximum of teleport distances for meeting participants. We will select the vertices v_1, …, v_X in such a way that the meeting cost is minimal possible.

The value of X depends on the coronavirus situation, and may vary from 1 to K. To prepare for the meeting in advance, write a program that, for each of the N meetings, finds the sum of the meeting costs for all possible values of X from 1 to K, inclusive.

Input

The first line of input contains two integers N and K: the number of vertices and the upper limit for X, respectively ($1 \le K \le N \le 3 \cdot 10^5$).

The following $N-1$ lines describe the tree. Each of these lines contains three integers, A_i, B_i, and C_i, telling that there is an edge between vertices A_i and B_i with weight C_i ($1 \le A_i, B_i, C_i \le N$). It is guaranteed that the resulting graph is a tree.

Output

Print N lines. On line i, print the sum of meeting costs of i-th meeting for all X from 1 to K, inclusive.

Examples

standard input	standard output
10 4	0
5 1 2	4
1 6 4	13
6 2 1	21
2 8 9	23
8 3 5	23
3 4 8	30
4 10 9	31
10 9 8	33
9 7 7	34
8 3	0
7 3 4	8
4 5 2	14
3 6 1	16
6 8 6	16
8 5 1	16
2 5 8	18
1 5 2	18

Problem C. Colorful Squares

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	1024 mebibytes

There are N points on a plane: P_1 (x_1, y_1), P_2 (x_2, y_2), $\ldots$, P_N (x_N, y_N). Each point has a color denoted by an integer from 1 to K.

Consider a square with edges parallel to coordinate axes. The square is *colorful* if it contains points of all K colors inside or on its border. It is allowed for the square to have edge length 0, thus covering just a single point.

Find a colorful square with the minimum side length, and print that length.

Input

The first line contains two integers N and K $(2 \le N \le 10^5, 2 \le K \le N)$.

Then N lines follow. The i-th of these lines contains three integers, x_i, y_i, and c_i: the coordinates of the i-th point and its color, respectively $(1 \le x_i, y_i \le 2.5 \cdot 10^5, 1 \le c_i \le K)$. You may assume that there is at least one point for each of k colors.

Output

Print one integer: the answer to the problem.

Examples

standard input	standard output

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

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

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

Problem E. Expected Distance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

Given are two integer sequences: $\{a_i\}$ of length $N-1$ and $\{c_i\}$ of length N. Let us build a tree T_N with N vertices in the following way:

- T_1 is a tree made up of only vertex 1.
- For $i > 1$, T_i connects vertex i to one of the vertices of T_{i-1}. The probability that vertex j will be chosen is $\frac{a_j}{a_1 + \cdots + a_{i-1}}$. The length of the added edge is then calculated as $c_i + c_j$.
- When T_N is built, the process stops.

You are given Q queries. Each query is a pair of vertices. For each query (u, v), calculate the expected distance between u and v in T_N.

Input

The first line of input contains two integers N and Q: the number of vertices and the number of queries, respectively ($2 \le N, Q \le 3 \cdot 10^5$).

The second line contains $N-1$ integers $a_1, a_2, \ldots, a_{N-1}$ ($1 \le a_i \le 2000$).

The third line contains N integers $c_1, c_2, \ldots, c_N$ ($1 \le c_i \le 2000$).

Each of the following Q lines describes one query and contains two integers u and v separated by a space: numbers of vertices to find the expected distance ($1 \le u, v \le N$).

Output

It can be proven that each answer is a rational number and can be written in the form $ans_i = \frac{p_i}{q_i}$, where p_i and q_i are coprime non-negative integers and $0 < q_i < 10^9 + 7$. For each query, print the integer $(p_i \cdot q_i^{-1}) \bmod (10^9 + 7)$.

Examples

standard input	standard output

5 7
1 1 1 1
1 2 4 8 16
1 3
2 5
4 3
1 5
3 3
4 5
1 2
	7
	666666697
	15
	666666697
	0
	333333366
	3

5 4
17 19 23 29
2 3 5 7 11
1 2
3 4
5 2
3 5
	5
	927495315
	106531441
	450222593

Problem F. Find the XOR

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There is a connected graph G consisting of N vertices and M weighted undirected edges. In G, the weight of the path is obtained by XORing the weights of all edges in the path. Note that it is allowed to walk along one edge multiple times, in this case, you are XORing the weight that number of times.

For vertices u and v, let $d(u, v)$ be the **maximum weight** of a path from u to v.

You need to answer Q queries of the following form:

Given l and r, for all i and j such as $l \le i < j \le r$, find the XOR of the values of $d(i, j)$.

Input

The first line contains three integers, N, M, and Q ($1 \le N, M, Q \le 10^5$).

Each of the following M lines contains three integers, u, v, and w, describing an edge of weight w connecting vertices u and v ($1 \le u, v \le N$, $0 \le w < 2^{30}$). Note that multiple edges and loops are **allowed** in this task.

Each of the following Q lines describes one query and contains two integers l and r ($1 \le l < r \le N$).

Output

For each query, print the answer on a separate line.

Example

standard input	standard output

```
8 10 7
1 2 662784558
3 2 195868257
3 4 294212653
4 5 299265014
6 5 72652580
6 7 29303370
7 8 183954825
2 1 752722885
5 3 197591314
8 4 877461873
4 8
5 7
6 7
2 3
7 8
3 4
2 7
```
	```
	0
	713437792
	738051848
	716356296
	736682272
	1003204975
	987493236
	```

Problem G. Generate The Array

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Consider an array A of length N. You are planning to do several queries: for a segment $[i, j]$ of the array, find the maximum value on that segment of the array. The query for the indices i and j will be done $Q_{i,j}$ times.

But the array is not given, and you are going to build it right now.

For each i from 1 to N, you can select one of K_i different values $V_{i,j}$ as the value of A_i, and the respective costs of choosing these values are $C_{i,j}$.

After all queries, your *score* will be the sum of the results of all the interval queries you are planning to do, minus the cost of choosing the values A_i. Find the maximum possible score that can be achieved.

Input

First line of the input contains one integer N ($1 \leq N \leq 300$).

Then N lines follow. The i-th of those lines contains integers from $Q_{i,i}$ to $Q_{i,N}$ ($0 \leq Q_{i,j} \leq 999$). The query for the maximum element in the array between A_i and A_j inclusively shall be performed exactly $Q_{i,j}$ times.

After that, the input describes possible values of A_i for each i from 1 to N. The i-th description has the following format:

- The first line contains a positive integer K_i: the number of possible values for A_i.
- Each of the following K_i lines contains two integers $V_{i,j}$ and $C_{i,j}$: a possible value and the cost of picking that value, respectively ($0 \leq V_{i,j} \leq 10^8$, $0 \leq C_{i,j} \leq 10^{13}$).

It is guaranteed that the sum of K_i is at most $3 \cdot 10^5$.

Output

Print one integer: the maximum possible score.

Examples

standard input	standard output

5
1 0 2 2 0
0 2 2 0
2 2 2
1 2
0
2
0 27
1 19
2
7 25
1 1
2
8 7
4 18
2
8 7
4 4
2
0 25
4 26
	78

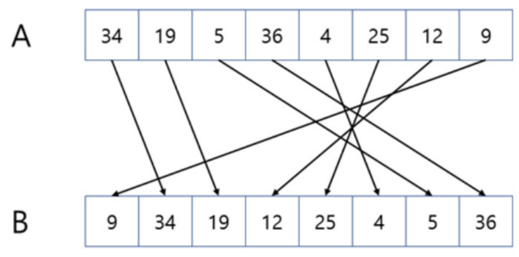
2
1 1
1
2
1 100
2 50
1
1 100
	-145

Problem I. Integer Array Shuffle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Given an integer array A of size N, the *shuffle* operation is defined as follows.

- Initially, you create an empty integer array B.
- Then, while A is not empty, you remove either the leftmost or rightmost element of A, and append the value to the right in B.
- If A is empty, replace A with B and stop.



If the shuffle operation is performed as shown in the picture above, the value of the array A is changed as follows:

$$(34, 19, 5, 36, 4, 25, 12, 9) \to (9, 34, 19, 12, 25, 4, 5, 36).$$

Let A_i be the i-th element of array A. When the condition "if $1 \le i < j \le N$, then $A_i \le A_j$" is established, it is said that array A increases monotonically.

Write a program that, given an integer array A of size N, calculates the minimum number of shuffle operations required to make the array A monotonically increasing.

Input

The first line of input contains one integer N, the number of elements in array A ($1 \le N \le 3 \cdot 10^5$).

The second line contains N integers $A_1, \ldots, A_N$: the initial values of elements of array A ($1 \le A_i \le 10^9$).

Output

Output the minimum number of shuffle operations required to make the array A monotonically increasing.

Examples

standard input	standard output
3 2 2 5	0
6 1 5 8 10 3 2	1

Problem J. Junkyeom's Contest

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Junkyeom and his friends Myung and Myeong are planning to hold a programming contest with one gold (first place), two silver (places 2 and 3), and four bronze medals (places 4, 5, 6, 7).

The sponsors gave N gift cards for the contest, i-th of them costs A_i. Each medalist shall be awarded exactly one card. Let P_i be the prize for the card awarded to the contestant taking i-th place. The distribution is considered *fair* if the following two inequalities are held:

$$P_1 \geq P_2 \geq P_3 \geq P_4 \geq P_5 \geq P_6 \geq P_7$$

and

$$P_1 < P_2 + P_3 < P_4 + P_5 + P_6 + P_7.$$

Given the values A_i, find out if a fair distribution of prizes exists. If it does, print the maximum possible sum of P_i for a fair distribution.

Input

The first line of input contains one integer N, the number of gift cards ($7 \leq N \leq 5 \cdot 10^5$).

The second line contains N integers A_i: the prizes for the cards ($1 \leq A_i \leq 2 \cdot 10^8$).

Output

If a fair distribution of prizes is impossible, print -1.

Otherwise print one integer: the maximum possible total prize of the fairly distributed gift cards.

Examples

standard input	standard output
7 1 2 3 4 5 6 7	-1
8 1 2 3 4 5 6 7 8	35
10 5 5 5 5 5 5 10 5 5 5	35

Problem M. Move To The Equality

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Take any four positive integers: a, b, c, d. Form four more, like this: $|a-b|$ $|b-c|$ $|c-d|$ $|d-a|$. Then, do it again with these four new numbers. And then again. And again. Eventually, all four integers will be the same.

Given a, b, c and d, figure out how quickly the sequence converges.

Input

There will be no more than 1100 several test cases in the input.

Each test case consists of four positive integers on a single line ($1 \le a, b, c, d \le 2 \cdot 10^9$) in that order. The input will end with a line with four zeros, which should not be processed.

Output

For each test case, print a single integer on its own line, indicating the number of steps until convergence.

Examples

standard input	standard output

1 3 5 9
4 3 2 1
1 1 1 1
0 0 0 0
	6
	4
	0

Problem N. Numbers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

A number n is called Special if it has a pair of factors, a and b, where $a \cdot b = n$, and together, a and b have exactly the same digits, in exactly the same quantities, as n. None of the numbers n, a or b can have leading zeros.

Here are some examples: $126 = 6 \cdot 21$, $10251 = 51 \cdot 201$, $702189 = 9 \cdot 78021$ Given an integer X, find the smallest Special number which is greater than or equal to X.

Input

There will be no more than 432 test cases in the input.

Each test case will consist of a single line containing a single integer X ($10 \le X \le 10^6$). The input will end with a line with a single 0.

Output

For each test case, output a single integer on its own line, which is the smallest Special number which is greater than or equal to X.

Examples

standard input	standard output
1 126 128 5555 0	126 126 153 6880

Problem O. Olympic Tournament

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Alex missed the figure skating olympic qualification that he wanted to attend. So now he wants to know the pairs of skaters whose dancing he missed. He had several photos from the warmup, so he chose one where all skaters are clearly visible and wrote down the coordinates of all N skaters (N is even).

Then Alex determined the pairs of skaters by the following algorithm: from not yet paired skaters he chooses two closest (to each other) skaters and assumes that they dance together as a pair. Should he find several pairs of skaters with the same minimum distance between skaters, he chooses lexicographically smallest pair (Alex enumerated skaters by integers from 1 to N, skaters are ordered inside a pair, one with lower number goes first). You are asked to help Alex to determine pairs of skaters.

Input

The first line of input contains even integer N ($2 \le N \le 300$). Each i-th line of the next N lines contains two integers — x and y coordinates of point, representing i-th skater. All points are distinct. All coordinates are less than 10^8 by absolute value.

Output

You should output $N/2$ lines. Each line must contain numbers of skaters in the corresponding pair. The first number in a line should be less than the second. Lines must be sorted in the lexicographically ascending order.

Examples

standard input	standard output

6
0 2
3 2
0 0
1 0
0 -2
2 -2
	1 2
	3 4
	5 6

4
0 0
1 1
0 1
1 0
	1 3
	2 4

Problem P. Product Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Rules of The Product game are simple. Master announces positive integer N. Player needs to calculate product of digits for each positive integer up to N and report the greatest product.

All answers should be known beforehand to allow interactive communication during a game. The Game Master asks you to write a program which having positive integer N will find correct answer.

Input

Input file contains one integer N ($1 \leq N \leq 2 \cdot 10^9$).

Output

Output file should contain greatest product for given N.

Examples

standard input	standard output
1	1
101090000	43046721
28994	10368
4876	2268
2789	1008

Problem Q. Quantum Robot

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 Mebibytes

Secret laboratory of Bytesburg has developed a new quantum robot which moves on stripe consisting of $n+1$ cells. Cells are numbered from 0 to n. The robot is located at cell with number 0; each other cell contains several viruses; as long as robot appears into that cell, viruses are destroyed.

The robot can do m single jumps (to adjacent cell) and k double jumps (over one cell). Additionally, $m + 2k = n$. All jumps are jumps forward. Write a program which finds sequence of jumps with highest number of viruses destroyed.

Input

The first line at the input contains 3 integers: n $(1 \le n \le 100)$, m $(0 \le m \le 100)$, k $(0 \le k \le 100)$. The second line contains n integers — number of viruses (up to 100) in corresponding cells of the stripe.

Output

The first line at the output should contain highest number of viruses found. The second line should contain $m+k+1$ integers — numbers of cells visited by the robot, starting from cell with number 0.

Example

standard input	standard output
5 1 2 5 2 7 3 1	13 0 1 3 5