Task 1: `Snail`

A snail is stuck at the bottom of a well which is of height H. Initially, the snail is at height 0 and it tries to get out. Well, being at the bottom (i.e., height 0) has its upsides. For instance, the snail can't slide down any further (cannot slide into negative height).

A day is made out of N phases. During each phase, the snail will either try to climb up a certain amount, or rest and slide down a certain amount. The snail knows how much it would move in each phase and this is denoted by P_i. If positive, this means that the snail will travel upwards, if negative, the snail will slide downwards (until height 0), if zero, the snail will maintain its height.

Find out the first day and phase such that the snail reaches a height of H to exit the well.

Warning: For C++ users, remember to use `long long` data type (and `%lld` format specifier for scanf and printf), for Java users, remember to use `long` data type.

Input format

Your program must read from standard input.

The first line of input contains two positive integers H, denoting the height of the well and N, denoting the number of phases per day.

The next line of input contains N signed integers (separated by spaces) describing the snail's routine: $P_0, P_1, P_2, \ldots, P_{N-1}$ where P_i denotes the amount the snail travels in phase i.

Output format

Your program must print to standard output.

Your program should print one line with two integers, separated by a space.

If the snail can reach the top of the well (i.e. height H) for the first time on the phase P of day D, your output should be D followed by P.

Otherwise, if the snail will always be stuck in the well, your output should be -1 followed by -1.

Subtasks

The maximum execution time on each instance is 1.0s.

For all subtasks, $1 \leq H \leq 10^{12}$, $-10^{12} \leq P_i \leq 10^{12}$ and $1 \leq N \leq 10\,000$.

Your program will be tested on sets of input instances as follows:

Subtask	Marks	Limits
1	11	$N = 1$
2	9	$P_i = P_0$ for all i.
3	25	$H \times N \leq 10\,000$
4	17	$P_i \geq 0$ for all i
5	38	No further restrictions

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
`3 1` `1`	`2 0`

Explanation

After phase 0 of day 0, the snail has climbed 1 unit.
After phase 0 of day 1, the snail has climbed 1 unit to a height of $1 + 1 = 2$ units.
After phase 0 of day 2, the snail has climbed 1 unit to a height of $1 + 2 = 3$ units and exits the well.

Sample Testcase 2

This testcase is valid for subtasks 1, 2, 3, 5.

Input	Output
5 1 -1	-1 -1

Explanation

The snail cannot climb upwards at all, hence, the snail is stuck in the well.

Sample Testcase 3

This testcase is valid for subtasks 3, 5.

Input	Output
5 2 4 -2	1 0

Explanation

After phase 0 of day 0, the snail has climbed 4 units.
After phase 1 of day 0, the snail has slid down 2 units to a height of 2 units.
In phase 0 of day 1, the snail climbs 3 more units, up to a height of $2 + 3 = 5$ units, and exits the well.

Task 2: `Knapsack`

There is a housewife who recently won a prize to "shop for free as long as your shopping basket is not full" in a department store.

This housewife is given a shopping basket that can carry a maximal weight of S kilograms.

There are N item types in the department store and the i-th item is worth V_i SGD, weighs W_i kilograms, and there are K_i copies (of exactly same value and weight) of such item i.

For example, there are $N = 3$ item types: meat, milk, and bread; of which there are: 1 pack of meat, 3 bottles of milk, and 4 loaves of bread (see the last sample test case).

What items should the housewife take to maximize the total value of the items in her shopping basket?

Input format

Your program must read from standard input.

The first line of input contains two positive integers, S and N.

The next N lines of input will each contain three integers, where the i-th line contains V_i, W_i and K_i, the value in SGD, weight in kilograms and number of the i-th item respectively.

Output format

Your program must print to standard output.

Your program should print one integer, representing the maximum total value in SGD of the items that this housewife can take while ensuring the total weight does not exceed S kilograms.

Subtasks

The maximum execution time on each instance is 1.0s.

For all subtasks, $1 \leq S \leq 2000$, $1 \leq V_i \leq 1000000$, $1 \leq W_i \leq S$.

Your program will be tested on sets of input instances as follows:

Subtask	Marks	Limits
1	12	$N = 1$
2	17	$1 \leq N \leq 100$, $K_i = 1$
3	20	$1 \leq N \leq 100$, $1 \leq K_i \leq 10$
4	24	$1 \leq N \leq 100$, $1 \leq K_i \leq 10^9$
5	27	$1 \leq N \leq 100000$, $1 \leq K_i \leq 10^9$

Sample Testcase 1

This testcase is valid for subtasks 2-5.

Input	Output
15 5 4 12 1 2 1 1 10 4 1 1 1 1 2 2 1	15

Explanation

The housewife can take one of items 2, 3, 4, 5 giving a total weight of $1 + 4 + 1 + 2 = 8$ and a total value of $2 + 10 + 1 + 2 = 15$.

Sample Testcase 2

This testcase is valid for subtasks 3-5.

Input	Output
20 3 5000 15 1 100 1 3 50 1 4	5400

Explanation

The housewife take one of item 1, three of item 2 and two of item 3 for a total weight of $15 \times 1 + 1 \times 3 + 1 \times 2 = 20$ and a total value of $5000 \times 1 + 100 \times 3 + 50 \times 2 = 5400$.

Task 3: `Island`

Squeaky the Mouse has just discovered an inhabited, circular island and is sailing back to his homeland to announce this new discovery. The island is circular and made up mostly of rocky, non-arable land. As such, saltwater fish is the main food source to the inhabitants, so they reside in N towns (numbered 1 to N) along the coast of the island.

To connect the towns, M junctions have been created on the interior of the island and some roads built to connect the towns and the junctions. To minimise construction cost, exactly $N + M - 1$ roads are built such that it is possible to travel by road between any two towns, and there is exactly one road ending at each town. In other words, the road network may be represented as a tree with N leaves (representing the N towns), M internal nodes (representing the M junctions), and $N + M - 1$ edges (representing the $N + M - 1$ roads).

Furthermore, every junction has at least three roads connected to it, roads do not meet other roads except at junctions, and there are no bridges or tunnels (they are expensive).

Here is an example map of the island, with 37 towns, 20 junctions, and 56 roads:

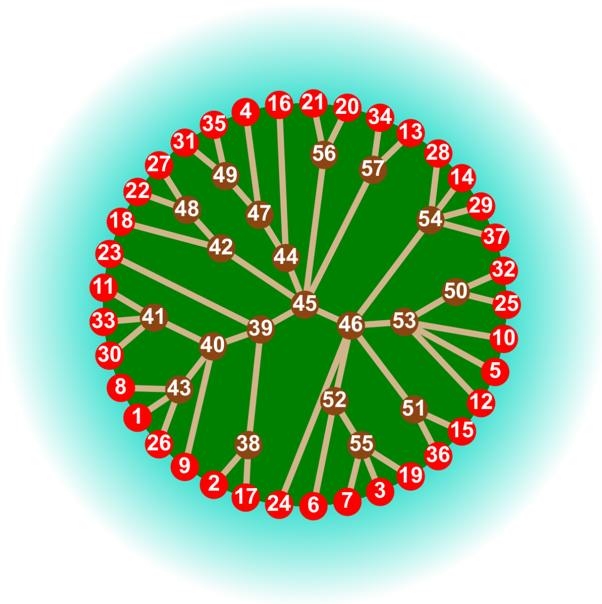

Figure 1: Example map of the island

This island is so intriguing that Squeaky is already making plans for his next trip, in a bigger vessel, where he will sail around the whole island, visiting the towns in the order that they are located along the coast. To do this, it is important to know the ordering of the towns along the coast.

Unfortunately, due to strong winds during the journey home, the maps that Squeaky had meticulously crafted were blown off his ship and are now forever lost to the depths of the ocean.

However, not all is lost. Squeaky had brought along a small journal in which he noted down the two endpoints of every road on the island. From this information, he hopes to find the possible orderings of towns along the circular coast, and has tasked you to find them for him. Note that since this island is circular, rotations of the same ordering are treated as equivalent (see example for more details).

To complete your task, you need to know the number of possible orderings, P, of towns around the coast. Since this number may be large, write this value as a product of factors raised to positive exponents (see output section for details).

Input format

Your program must read from standard input.

The first line of input contains two positive integers N and M, where N is the number of towns and M is the number of junctions.

The next $N + M - 1$ lines of input will each contain two integers u and v. This means that the town or junction numbered u has a direct road to the town or junction numbered v. (If $u \leq N$, it represents a town; otherwise, it is a junction. Similarly for v.)

The input is guaranteed to fulfill all constraints stated above, and at least one valid road network is guaranteed to satisfy the input.

Output format

Your program must write to standard output.

Your program must output K lines describing the number of possible orderings, P, of towns around the coast.

The i^{th} line must contain exactly two integers, a_i and b_i.

Your output must satisfy:

- $P = a_1^{b_1} a_2^{b_2} a_3^{b_3} \cdots a_K^{b_K}$ (or equivalently, $P = \prod_{i=1}^{K} a_i^{b_i}$)
- $1 \leq a_i, b_i \leq 10^{18}$
- $0 \leq K \leq 10^6$

It is guaranteed that the number of possible orderings is expressible in this form.

Subtasks

The maximum execution time on each instance is 1.0s.

For all subtasks, $N \geq 2$ and $M \geq 0$.

Your program will be tested on sets of input instances as follows:

Subtask	Marks	Limits
1	7	$N + M \leq 200\,000$, $M \leq 1$
2	20	$N + M \leq 200\,000$, $N \leq 10$
3	31	$N + M \leq 1\,000$
4	42	$N + M \leq 200\,000$

Sample Testcase 1

This testcase is valid for Subtasks 2-4.

Input	Output
5 2 1 7 3 7 6 2 7 4 6 7 5 6	3 1 4 1

Explanation

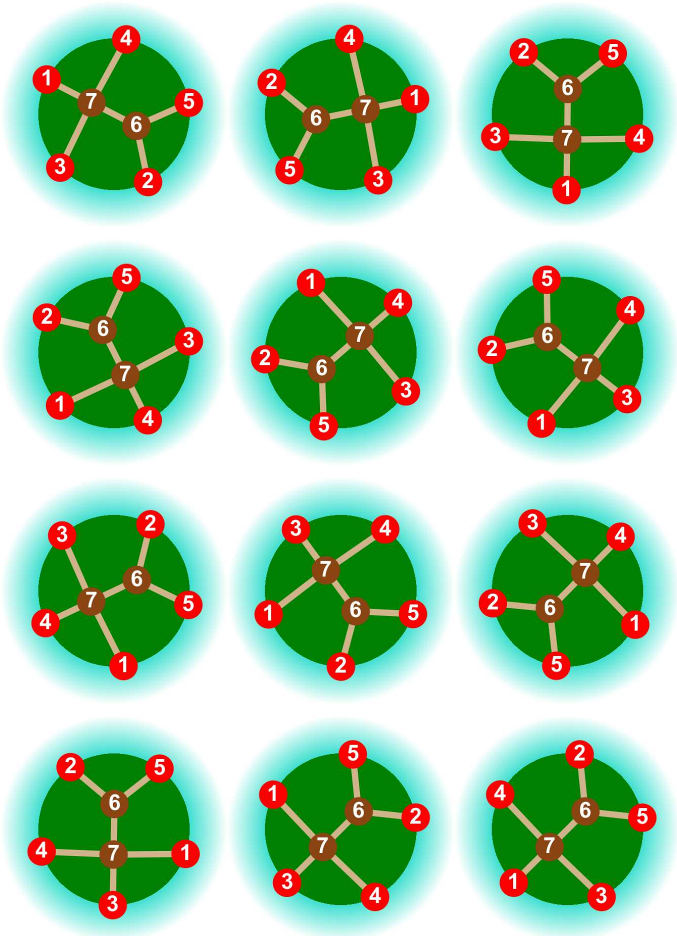

Figure 2: Possible maps of the island for sample testcase 1

The 12 distinct configurations above are the only possible orderings of the towns around the coast. As such, the number of possible orderings, P, is 12. Since $3^1 \times 4^1 = 12$, the output is correct. Note that there are other ways to represent 12 in the output, and those outputs will also be judged as correct.

Sample Testcase 2

This testcase is valid for all subtasks.

Input	Output
5 1 6 1 6 2 6 3 6 4 6 5	3 1 2 3

Explanation

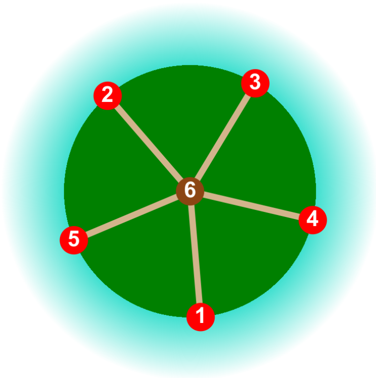

Figure 3: A possible map for sample testcase 2

There are 24 possible orderings of the island in this input. As such, the number of possible orderings, P, is 24. Since $3^1 \times 2^3 = 24$, the output is correct. Note that there are other ways to represent 24 in the output, and those outputs will also be judged as correct.

Sample Testcase 3

This testcase is valid for Subtasks 2-4.

Input	Output

`6 3`
`7 1`
`7 2`
`8 3`
`8 4`
`9 5`
`9 6`
`7 8`
`9 8`
	`24 1`

Explanation

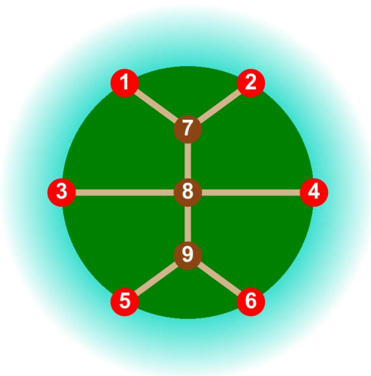

Figure 4: A possible map for sample testcase 3

There are 24 possible orderings of the island in this input. As such, the number of possible orderings, P, is 24. Since $24^1 = 24$, the output is correct. Note that there are other ways to represent 24 in the output, and those outputs will also be judged as correct.