



Task 4: Firefighting (firefighting)

A terrible fire swept through Mouseopolis, the capital of the Great Kingdom of Mouseland. Many residents of Mouseopolis suffered great losses as the fire razed through large chunks of the town. As the residents of Mouseopolis picked themselves up and slowly put their livelihoods back together after the fire, Squeaky — the benevolent king of Mouseland — vowed to prevent such a horrifying disaster from ever occurring again.

Mouseland consists of N towns, numbered from 1 to N . There are $N - 1$ bidirectional roads linking towns — each road is a direct link between some pair of towns. These roads may have varying lengths. The road network is designed such that it is possible to travel between any two towns using some sequence of roads. To quickly put out any fires, Squeaky has decided that fire stations should be built in some of the towns in Mouseland. These fire stations would then be able to dispatch fire trucks to mount an effective response to fires in any town.

Squeaky's advisors have determined that an effective response can be mounted only if the nearest fire station is no more than K kilometres away from the fire, as fire trucks would only then be able to reach the fire before the fire gets out of control. However, fire stations are expensive to maintain, so Squeaky would like to minimise the number of fire stations that need to be built.

Your task is to determine which towns fire stations should be built in, such that the required number of fire stations is minimised.

Note that fire stations may only be built in towns, and fires may only occur in towns.

Input

Your program must read from standard input.

The first line contains two integers, N and K , defined above.

$N - 1$ lines follow, describing the $N - 1$ bidirectional roads. Each of these lines contain three integers, A_i , B_i , and D_i , meaning that towns A_i and B_i are linked by a bidirectional road of length D_i kilometres.

Output

Your program must print to standard output.

The output should contain exactly two lines.

The first line should contain a single integer, F , the number of fire stations required.

The second line should contain exactly F distinct integers, specifying the towns in which fire stations should be built. This list may be printed in any order. If there are multiple possible ways, all of them will be accepted.



Note: If you receive the “Output is invalid” verdict, it means that you did not follow the required output format exactly. These are errors that can be detected with no information apart from the value of N (i.e. without knowing the road network configuration). For example: the F you printed is not an integer between 1 and N , or there are not exactly F integers that follow, or that they are not distinct integers between 1 and N .

Implementation Note

As the input and output lengths for subtasks 1, 4, and 6 may be very large, you are recommended to use C++ or Java with fast input and output routines to solve this problem.

C++ and Java source files containing fast input/output templates have been provided in the attachment. You are strongly recommended to use these templates.

If you are implementing your solution in Java, please name your file `Firefighting.java` and place your `main` function inside `class Firefighting`.

Subtasks

The maximum execution time on each instance is 3.0s, and the maximum memory usage on each instance is 1GiB. For all testcases, the input will satisfy the following bounds:

- $1 \leq N \leq 3 \times 10^5$
- $1 \leq K \leq 10^{15}$
- $1 \leq A_i, B_i \leq N$ for all $1 \leq i \leq N - 1$
- $1 \leq D_i \leq 10^9$ for all $1 \leq i \leq N - 1$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	3	$K \leq 20$, $D_i \geq 30$ for all $1 \leq i \leq N - 1$
2	5	$N \leq 17, K \leq 17$, $D_i = 1$ for all $1 \leq i \leq N - 1$,
3	9	$N \leq 17, K \leq 10^6$, $D_i \leq 10^4$ for all $1 \leq i \leq N - 1$,
4	19	$K \leq 30$, $D_i \geq 20$ for all $1 \leq i \leq N - 1$
5	26	$N \leq 3 \times 10^3$
6	38	-



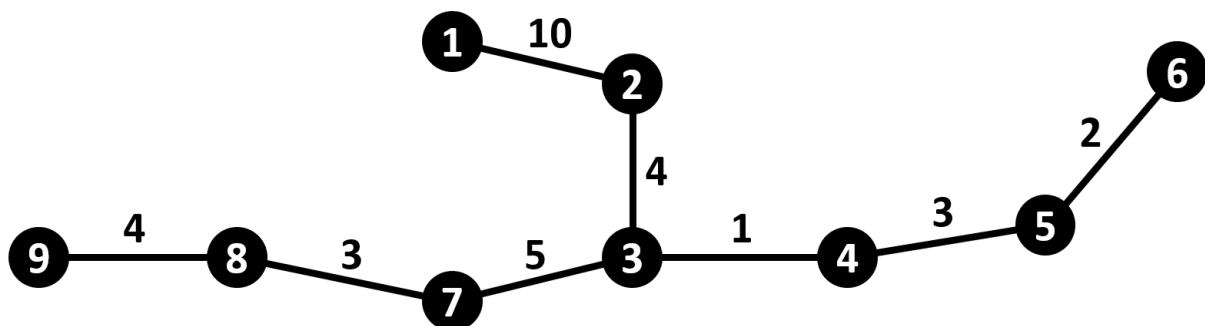
Sample Testcase 1

This testcase is valid for subtasks 3, 5, and 6.

Input	Output
9 4	4
1 2 10	1 8 3 6
2 3 4	
8 9 4	
8 7 3	
7 3 5	
3 4 1	
4 5 3	
5 6 2	

Sample Testcase 1 Explanation

Mouseland looks like this:



The minimum number of fire stations required is 4, and we can build fire stations at towns 1, 3, 6, and 8. After building fire stations at those towns, every town will be at most 4 kilometres away from the nearest fire station. Observe that the list of towns may be printed in any order.

Note that there are other valid solutions — for example, we can instead build fire stations at towns 1, 2, 5, and 8. All valid solutions will be accepted.



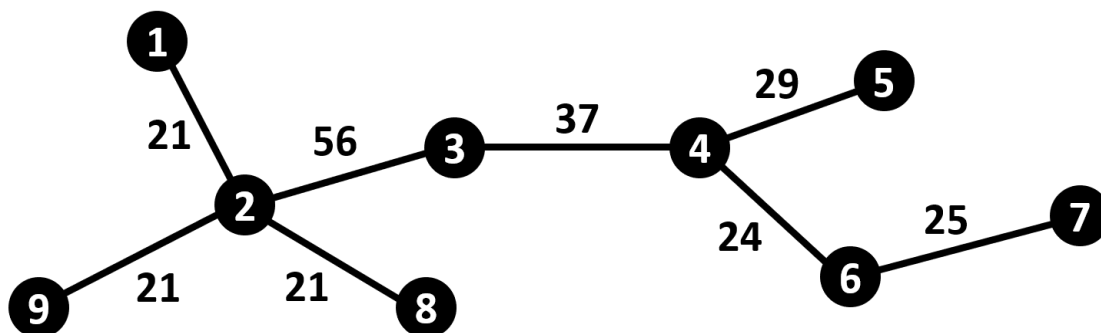
Sample Testcase 2

This testcase is valid for subtasks 3, 4, 5, and 6.

Input	Output
9 26	4
2 1 21	5 6 2 3
2 9 21	
2 8 21	
2 3 56	
3 4 37	
4 5 29	
7 6 25	
6 4 24	

Sample Testcase 2 Explanation

Mouseland looks like this:



The minimum number of fire stations required is 4, and we can build fire stations at towns 2, 3, 5 and 6. After building fire stations at those towns, every town will be at most 26 kilometres away from the nearest fire station. The list of towns may be printed in any order.

It happens that for this test case there are no other set of 4 towns that will satisfy the requirements.



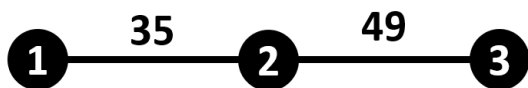
Sample Testcase 3

This testcase is valid for subtasks 1, 3, 4, 5, and 6.

Input	Output
3 18 1 2 35 2 3 49	3 1 2 3

Sample Testcase 3 Explanation

Mouseland looks like this:



The minimum number of fire stations required is 3 — we have to build a fire station at every town.



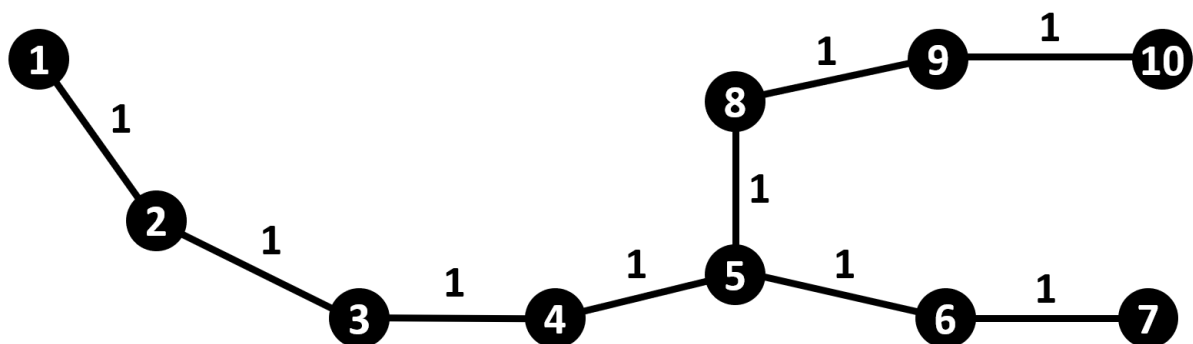
Sample Testcase 4

This testcase is valid for subtasks 2, 3, 5, and 6.

Input	Output
10 3	2
10 9 1	4 10
9 8 1	
8 5 1	
1 2 1	
2 3 1	
5 4 1	
3 4 1	
5 6 1	
6 7 1	

Sample Testcase 4 Explanation

Mouseland looks like this:



The minimum number of fire stations required is 2, and we can build fire stations at towns 4 and 10. After building fire stations at those towns, every town will be at most 3 kilometres away from the nearest fire station. The list of towns may be printed in any order.

Note that there are other valid solutions — for example, we can instead build fire stations at towns 3 and 5. All valid solutions will be accepted.