Paired roads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

In a certain country, there are n cities, between which no roads have been built yet. The i-th city has a population of w_i people. There are also $n-1$ roads that can be built. The i-th road, when built, will connect cities u_i and v_i, and its construction cost is s_i.

It is known that if all the roads are built, then from every city it will be possible to reach any other city using only these roads. In other words, the roads form a tree.

During each of the following k days, the following will happen: on the i-th day, a city c_i is chosen, as well as two distinct roads that have not been built yet, connecting this city to some other cities. After that, these two roads are constructed, and the price paid is equal to the sum of the construction costs of these two roads. The city c_i is then considered *central* on day i.

In k days, each city that is considered *central* at least once will generate profit equal to the number of people living in it.

We define *benefit* as the difference between the profit brought by the cities and the total cost of the constructed roads. Find the maximum possible *benefit*.

Input

The first line contains three integers n, k, and t ($3 \le n \le 200\,000$, $1 \le k \le \frac{n-1}{2}$, $0 \le t \le 1$) — the number of cities, the number of pairs of roads to be built, and integer t, which is equal to 1 if it's required to find which roads to build, and 0 otherwise.

The second line contains n integers w_1, w_2, ..., w_n ($1 \le w_i \le 10^8$) — population of each city.

Each of the next $n-1$ lines contains three integers u_i, v_i, and s_i ($1 \le u_i, v_i \le n$, $1 \le s_i \le 10^8$) — the cities connected by the i-th road and the cost of building it.

It is guaranteed that the roads form a tree, and that it is possible to build k pairs of roads that satisfy the given condition.

Output

In the first line, print a single integer — the maximum *benefit* that can be obtained after building exactly k pairs of roads.

If $t = 1$, then in the next k lines, print k triples of numbers c_i, x_i, and y_i ($1 \le c_i, x_i, y_i \le n$) — the description of the pairs of roads that need to be built. Such a triple represents a pair of roads (c_i, x_i) and (c_i, y_i).

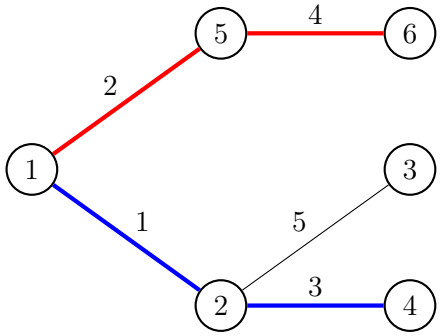
If there are multiple ways to achieve the maximum *benefit*, output any of them.

Examples

standard input	standard output
6 2 1 1 2 3 4 5 6 1 2 1 2 3 5 2 4 3 1 5 2 5 6 4	-3 5 6 1 2 4 1
8 3 0 4 5 1 2 3 1 3 5 2 1 15 7 1 5 4 8 1 8 5 2 7 8 1 6 7 5 3 7 7	-13

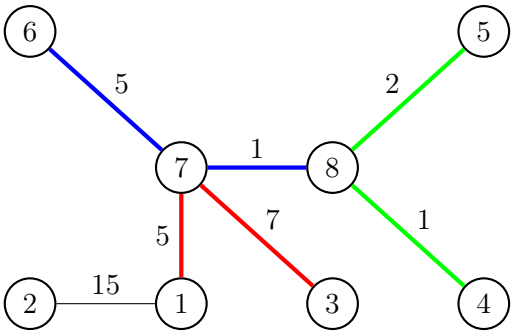
Note

In the first test from the problem statement, it is optimal to construct the following roads (roads from the same pair are marked with the same color):



The total cost of constructing these roads is equal to $2+4+1+3=10$. After construction, cities 2 and 5 will generate profits of 2 and 5 respectively. Thus, the *benefit* is $2+5-10=-3$. It can be shown that it is not possible to obtain a better answer.

In the second test from the problem statement, it is optimal to construct the following roads:



The total cost of constructing these roads is equal to $5+1+2+1+5+7=21$. After construction, cities 7 and 8 will generate profits of 3 and 5 respectively. Thus, the *benefit* is $3+5-21=-13$. It can be shown that it is not possible to obtain a better answer.

Scoring

The tests for this problem consist of 7 groups. Points for each group are awarded only if all the tests in that group and some tests from the previous groups pass. **Offline-testing** means that the results of testing your solution on this group will only be available after the competition ends.

Group	Score	Additional constaints		Required groups	Comment
		n	t		
0	0	–	–	–	Samples.
1	13	$n \leq 200$	$t = 0$	–	
2	17	$n \leq 1500$	$t = 0$	1	
3	12	$n \leq 1500$	–	0 – 2	
4	19	–	$t = 0$	–	$u_i = i$, $v_i = i + 1$
5	11	–	–	4	$u_i = i$, $v_i = i + 1$
6	15	–	$t = 0$	1, 2, 4	**Offline-testing.**
7	13	–	–	0 – 6	**Offline-testing.**