Tennis

This year, Athens is preparing to host a tennis event, with no real winner, to support various non-profit organizations. In order to raise as much money as possible, the organizers are planning to invite famous tennis players from all over the world.

The organizers decided that the event will consist of K matches, each between two different tennis players, and they asked your help to find which $2K$ tennis players to invite to maximize the number of tickets that will be sold in the event.

Unfortunately, you are not that much of a tennis fan yourself. So you searched the web and found a report from a famous international tennis tournament with $N \geq 2K$ players, that was held recently and lasted $N-1$ days. Each day a single match was held; the winner advanced and was eligible to play during the following days, whereas the loser left the tournament and did not play in any of the subsequent matches. For each of the $N-1$ matches, the report states which pair of players (x_i, y_i) participated in the match and how many people w_i attended. Since, you prefer not to rely on pure luck, you are only willing to select which players to invite based on the $N-1$ pairs you have some information about. However, to simplify your analysis, you make the assumption that selecting each of these pairs (x_i, y_i) will result in the same number of people w_i attending the event.

You are asked to write a program that computes the maximum number of tickets that can be sold in the event.

Input

Your program must read from the standard input.

The first line will contain two space-separated integers N and K: the number of players in the past tournament and the number of matches that you have to plan.

Each of the following $N-1$ lines will contain three space-separated integers x_i, y_i and w_i: the first player of the match, the second player of the match and the number of people that attended the match, which result in the same number of tickets being sold. Players will be numbered 1 to N.

Note that the matches will be provided in arbitrary (and not chronological) order.

Output

Your program must print a single line to the standard output, consisting of a single integer: the maximum number of tickets that can be sold in the event. If you don't have enough information to select exactly K pairs, you will have to output -1 as your answer.

Constraints

- $2 \leq 2K \leq N \leq 10^6$
- $1 \leq x_i < y_i \leq N$
- $1 \leq w_i \leq 10^6$
- Time and memory limit: See the CMS.

Subtasks

- Subtask 1 (11 points): $N \leq 10^3, \quad K \leq 100$
  Furthermore, in the past tournament, no player participated in more than 3 matches.
- Subtask 2 (24 points): $N \leq 10^4, \quad K \leq 100$
- Subtask 3 (27 points): $N \leq 1.5 \cdot 10^4, \quad K \leq 7 \cdot 10^3$
- Subtask 4 (38 points): No further constraints.

Example 1

Input

```
6 3
1 2 9
2 3 6
3 4 6
4 5 9
5 6 2
```

Output

```
17
```

Explanation

You select the pairs (1, 2), (3, 4) and (5, 6), as this is the only way to select exactly 3 valid pairs given your constraints. Selecting pairs (1, 2) and (4, 5) would result in more tickets being sold, but there would be no way for you to select a third match.

Example 2

Input

```
7 3
1 2 8
1 3 15
2 4 3
2 7 2
3 6 10
4 5 6
```

Output

```
24
```

Explanation

In this scenario, there are multiple ways to select exactly 3 pairs, but you select the pairs (1, 2), (3, 6) and (4, 5) which will result in the maximum amount of tickets being sold.

Example 3

Input

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

Output

```
-1
```

Explanation

In this scenario, there is no valid way to select more than 2 pairs of players to invite, since at least one of players 1 and 3 participated in all of the matches.