1. Bazaar

Next week is the grand Christmas market of the gorge, and you do not want to miss the opportunity to buy military equipment, so that you appear fully prepared for the battle of January. The equipment consists of an armor, a sword and a shield. We symbolize these objects with A for armor, B for sword and C for shield. There are 3 dealers who sell military equipment (which, for convenience, we denote by 1, 2 and 3 respectively). Each merchant can sell one or more items of any type.

To be sure that you will be able to fight every day, you have calculated that you need at least N complete sets of equipment (ie at least N items from each of the types A, B and C). But items sold by different merchants may not be compatible with each other. So a complete set of equipment counts every triad of items A, B and C purchased from the same dealer (ie you cannot complete a set of equipment by buying armor from dealer 1, and a sword and shield from dealer 2).

Each merchant announces at the beginning of the bazaar his offers, ie how many items of each kind he sells and at what price. Usually the same dealer sells many different (foreign to each other) sets of the same item at a different price (eg, dealer 1 can sell 3 pieces of armor, see type 1A items at a price of 100 and other 4 pieces of armor at a price of 800 ). The items mentioned in the offers are necessarily sold all together, as a single set (ie if you need to buy one or more armors from dealer 1, you must buy either the 3 1A armors at the price of 100, or the 4 1A armors at the price of 800, or all 7 armors together at the price of 900).

So you want to calculate the minimum amount you need to spend to buy N complete sets of military equipment.

Initially, your program will read from the standard input two positive integers N and M that represent the number of complete sets of military equipment you want to buy and the number of merchant offers, respectively. Each of the following M lines will contain an expression of the form: xy AP, where x \ in {1,2,3} denotes the merchant selling the item y \ in {A, B, C}, while A indicates the number of items of type xy and P the total value of A items (sold as a whole).

Output data: Your program must print in the standard output (on the first line) an integer that corresponds to the minimum amount you need to spend to purchase N complete sets of military equipment. In case the purchase of N complete set of equipment is not possible, the output must be -1. Explanation of 1st example: We buy 5 complete sets of equipment from dealer 1 with a total cost of 100 + 800 + 125 + 375 = 1400 and another 5 complete sets of equipment from dealer 2 with a total cost of 500 + 900 + 400 = 1800.

Restrictions 0 <= N <= 5000

1 <= M <= 1500

1 <= P <= 10 ^ 6

1 <= A <= 10

## 2. Chem

In a chemical laboratory, there are N different substances that are hazardous experimental waste and must be placed in K metal bottles to be safely transported to a special area outside the laboratory. The substances are numbered from 1 to N and, for safety reasons, should be placed in bottles in this order with the total amount of each substance in a single bottle. The bottles are large enough and the total amount of each substance small enough so that there are no capacity problems (ie even all the substances could fit in the same bottle). However, there is a risk of a chemical reaction between the substances in the same bottle, in which case significant amounts of energy are released. Specifically, for each pair of substances i and j contained in the same flask, the chemical reaction between them produces an energy equal to A [i, j] units. Based on the above, the procedure followed by the laboratory managers for the packaging of the substances is as follows: The first t1 substances in a row are placed in the first bottle, the next t2 substances in the second bottle, and so on, until place all substances in the K bottles. Thus, the energy that can be produced by the chemical reaction of the substances in the first flask is $\sum_{1 <= i < j <= t_1} A [i, j]$, for the second flask it is $\sum_{t_1 + 1 <= i < j <= t2} A [i, j]$, and so on. The total energy that could be produced by the chemical reaction of the substances in all K bottles is the sum of the above quantities. For safety reasons during the transport of the substances, the laboratory managers want to determine the ratios t1, t2, ..., t (K-1) of the substances where the bottle will be changed, so that the total energy that could be released from all bottles to be as small as possible. So you are asked to write a program for this purpose.

Input Data: Initially, the program will read from the standard input two positive integers N and K that represent the number of substances and the number of bottles. The program will then read N-1 lines, the i-th of which will contain N-i integers separated by spaces. The j-th integer of the i-th line corresponds to the energy A [i, j + i] (array A is symmetric about the diagonal, i.e. A [i, j] = A [j, i] for every 1 <= i < j <= N, and the diagonal has zero elements, i.e. A [i, i] = 0 for every 1 <= i <= N).

Output Data: The program must print in the standard output (on the first line) an integer corresponding to the minimum amount of energy that can be released.

Restrictions 0 <= A [i, j] <= 99 1 <= K <= 500 K <= N <= 1500

Bonus: some files with 1 <= K <= 700 and K <= N <= 2500

Execution time limit: 1 sec. Memory limit: 64 MB.