The World of Fragrance

Input File: *input_k.txt*
Output File: *output_k.txt*

Output Only Task

After some disappointing job experiences in the world of finance, you have decided to put your skills to a better use in the field of perfumery, where you now help to create the most exquisite perfumes. Creating new scents is an art that involes trying many different combinations of ingredients, and then using the perfumer's sense of smell to rate each new fragrance obtained.

Your new job is to help well trained perfumers save time, by generating only combinations of ingredients that are likely to smell good. You have collected data by asking the best noses in the world to rate various combinations of two ingredients at a time. Your theory is that if every pair of ingredients in a perfume smells good, then the perfume is likely to smell good overall.

You are provided with a total of N ingredients (numbered from 1 to N), and a list of P pairs of distinct ingredients with their corresponding ratings. A positive rating for a pair indicates a good smell, whereas a negative rating indicates a bad smell. Pairs of ingredients that have never been tested by experts are not listed and should be considered neutral, with a rating of 0.

The perfumers wish to create a new scent consisting of precisely K ingredients. Your task is to select a set of K distinct ingredients so that the sum of the ratings of every possible pair within this set is as high as possible. Note that you are not necessarily required to find the optimal solution. See the *Scoring* section for more details.

Constraints

- $2 \leq N \leq 1000$, where N is the number of ingredients available.

- $1 \leq K \leq 20$, where K is the number of distinct ingredients you need to include in your set.

- $1 \leq P \leq 100\,000$, where P is the number of pairs of ingredients that have been rated by experts.

- $-1000 \leq R_i \leq 1000$, where R_i is the rating of a given pair of ingredients.

Input

You are given ten input files, named `input_M.txt` ($1 \leq M \leq 10$).

The first line of each input file will contain the integers N K P, as described above. Each of the P remaining lines of the input file will contain three integers, A_i B_i R_i, indicating that the pair of ingredients (A_i, B_i) has been given the rating of R_i by the experts. It is guaranteed that $A_i \neq B_i$ for every such pair, and that no pair will be listed more than once.

Output

For each input file `input_M.txt`, you are required to produce a corresponding output file `output_M.txt` that describes your set of ingredients.

This output file should consist of precisely $K + 1$ lines. The first line must contain a single integer, the total of the ratings for all possible pairs of ingredients within your set. The next K lines must each contain a single integer describing one of the K distinct ingredients in your set.

Sample Input

```
5 3 7
1 2 12
1 3 10
1 5 -3
2 4 -2
2 5 -8
3 5 17
4 5 5
```

Sample Output

```
24
1
3
5
```

Explanation

In this example, you have five ingredients at your disposal, and must create a perfume using three of them. Seven pairs of these ingredients have been rated by experts, with ratings ranging -8 to 17.

The best possible set for this example is the combination of ingredients 1, 3 and 5. All three possible pairs within this set have been rated by experts: $(1, 3)$ with a rating of 10, $(1, 5)$ with a rating of -3, and $(3, 5)$ with a rating of 17, giving a total rating of $10 - 3 + 17 = 24$.

Scoring

There is no particular "best solution" that you are required to achieve. Instead, your score will be determined relative to the other contestants whom you are competing against (as well as the judges' solution). It is guaranteed that there will be at least one solution with a positive overall rating for each input scenario.

For each input scenario, the contestant who achieves the largest overall rating will be identified. Suppose that this contestant achieves a perfume with overall rating r. Your score for this input scenario will then be:

- 100% if your program finds a solution also with overall rating r;

- at least 10% if your program finds any valid solution;

- 0% if your program generates an incorrect solution (e.g. you incorrectly calculate the overall rating of your set);

- otherwise, determined by a formula relating your score against the best rating for the input scenario, as described below.

$$\text{Your score} = 10 + 90 \times \left(\frac{\text{your result}}{\text{best result}}\right)^5$$

For example, consider an input scenario for which the best solution finds a set with overall rating 40. Then the scoring scale for a *correct* solution would be as follows:

Overall rating	0	5	10	15	20	25	30	35	40
Score	10%	10%	10%	10%	12%	18%	31%	56%	100%

Submitting for an Output Only Task

To submit your output files, you must do the following:

- Create a zip file containing all of your output files. You may include as many or as few output files as you like (i.e., you do not need a solution for every input scenario). On a GNU/Linux system you can use a command like the following:

  ```
  zip mysolutions.zip output_*.out
  ```

  On Windows systems you can create a zip file by selecting *File → New → Compressed (zipped) Folder* from within Windows Explorer, and then you can copy your output files into this new zip file.

- Submit this zip file to the contest website, in the same way that you would submit an ordinary solution.