CLOUD ADVENTURE

Today, the software industry moves faster than ever. We want to watch our favourite TV shows with no lag or downtime, to use our online service anytime day or night, and work from anywhere. How can we do this? With cutting-edge Cloud Computing. Which is where Reply comes in. We help companies around the world implement the best cloud computing services for their specific projects. Want to help us? Then solve the problem below and try to find the best mix of providers to maximise the efficiency of our projects.

There are several different cloud service providers. Each one offers different regional data centres, and each data centre offers a number of different services. Each datacentre has a limited pool of each resource that customers can buy in packages composed of a fixed number of units per service. For example, if a package is composed of 25 units of a given service and you need 110 of those, you'll have to buy 5 x 25 = 125 total units. You can also mix-and-match packages from different providers to get the total number of resources you need.

Our client gave us a list of this year's projects, each one characterised in the following ways:

- Target country of software system users
- Number of units needed to operate a project for one day, for each resource
- The service level agreement (SLA) penalty, i.e. a fee to pay if our solution doesn't get the minimum amount of resources needed.

We also have statistical data about the average latency between each country and each provider's regional datacentre.

Our company evaluates project success by measuring different KPIs – the most important of them being the 'overall quality score', T_p. That's a 'lower-the-better' value and calculated using the formula:

$$T_p = \frac{Average\ project\ latency}{Overall\ availability\ index} \times (total\ operational\ project\ cost)$$

We calculate each factor as follows:

- **Average project latency**:
  For each provider region 'r' from which we bought resources for that project, we calculate the total number of computational units bought, multiplying the number of packages purchased by the number of service units in the package, regardless of their type. Let's call this amount 'U_r'. The latency statistic between the country of the project and each used provider's region is 'L_r'.
  The Average Project Latency is $\frac{\sum_r L_r \times U_r}{\sum_r U_r}$
  For example, if we buy for an Italian project:
  1 package from Amazon - Seattle containing 1 unit of cpu and 2 units of ram
  3 package from Microsoft - Redmond containing 5 units of cpu and 3 units of ram

  And there are the following latencies:

  Amazon-Italy 125

  Microsoft-Italy 130

  The Average Project Latency is:

  numerator: 125* [1*(1+2)] + 130*[3*(5+3)]

  denominator: 1*(1+2) + 3*(5+3)

  When $U_r = 0$, then we assume the Average Project Latency is 0.

- **Overall availability index**:
  For each service type, we call 'q_i' the number of units bought from a given region of a certain provider. For instance, if we assume the total requested quantity for a certain service type is 100, and we decide to buy 30 units from Provider X/Region Y and 70 units from Provider M/Region N, then we'll have $q_1 = 30, q_2 = 70$.
  We then calculate the availability index $'a_s'$ for a given service type 's' using the formula:
  $$a_s = \frac{(\sum q_i)^2}{\sum(q_i{}^2)}$$
  When no resources have been taken from a service type (meaning both $\sum q_i = 0$ and $\sum(q_i{}^2) = 0$), then we assume $a_s = 0$.
  We define the 'Overall availability index' as the average of all service availability indexes a_s of the project.

- **Operational Project Cost**:
  For each project resource, we have to buy the necessary amount from one or more providers and/or regions. Each provider sells its resources in packages of minimum buyable units, and bills a total of '*package fee x packages*' for their consumption. Let's define the '*total operational project cost*' as the sum of those costs.

When Overall Availability Index is 0, then we assume $T_p = 0$.

Our organisation is fined 'F_p' if a project doesn't meet its SLA, due to partial allocation of the service units it needs.

For each service type, we call 'F_s' the fine related to the service, calculated as:

$$F_s = (base\ project\ penalty) \times \frac{(units\ needed) - min(units\ needed,\ sum\ of\ allocated\ units)}{(units\ needed)}$$

When the units needed for a service are 0, we assume $F_s = 0$.

We define F_p as the average all of the service fines F_s of the project.

Your final score S will be:

$$S = \sum_p \frac{10^9}{T_p + F_p}$$

Input

Formal Description

The input file is a basic ASCII text file. Each line contains a different information separated by a white space.

On the initial line, you'll find the number of providers V, the number of services S, the number of countries C, and the number of projects P.

The next line specifies the S service names.

Then, the following line will list the names of all the C countries for the projects.

V blocks follow, each representing one provider. Each block contains several lines.

The first line of each block contains the name of the provider and the number of regions R_v it operates.

The next lines describe the regions. Each region occupies three lines:

- The first line contains the region name.
- The second line contains an integer representing the total number of service unit packages available in that region. Then there's a floating-point number representing the cost of a single package. S integer numbers u_i follows, representing the number of units of the i-esim service available in each package.
- The third line comprises C integer numbers, each one representing the latency between that region and the i-esim country.

Then, there come P lines describing the projects. Each one includes an integer representing the base penalty applied for violating the SLA. This is followed by a string displaying the country that will use the project, and finally by a list of S integer values corresponding to the amount of unit needed for each service type.

Structure

[V] [S] [C] [P]
[service 1] [service 2] ... [service S]
[country 1] [country 2] … [country C]
[provider 1] [num. of regions R1]
[region 1]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]
[region 2]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]

...

[region R1]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]
[provider 2] [num. of regions R2]
[region 1]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]
[region 2]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]

...

[region R2]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]

...

[provider V] [num. of regions RV]
[region 1]

[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]
[region 2]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]

...

[region RV]
[available packages] [package unit cost] [units of service per package 1] ... 0 ... [units of service per package S]
[latency 1] [latency 2] … [latency C]
[penalty] [country] [units 1] ... [units S]
[penalty] [country] [units 1] ... [units S]

...

[penalty] [country] [units 1] ... [units S]

Constraints

$1 \leq V \leq 20$
$1 \leq S \leq 500$
$1 \leq R_v \leq 100$
$1 \leq C \leq 20$
$1 \leq P \leq 100000$

Output
For each project, output a line describing how many packages you intend to buy from each provider. Assume the order of lines in the output file matches the order of the projects in the input file – i.e. the first line in the output file satisfies the first program in the input file. For each service needed, you can opt to buy the required number of service units made up of packages from different providers.

Each line is composed of a sequence of declarations, each one related to a single provider. Each declaration is built from the following information, separated by a single ASCII whitespace: the zero-based index of the provider in the input file, the zero-based index of the region inside the list of that provider, the number of packages you intend to acquire from that provider.

You can allocate more service units than the one requested by each project. If you don't intend to buy any packages from a particular provider and region, omit that declaration completely. If you intend to leave a project completely unsatisfied, output an empty line for it.

The output file must be a single, ASCII encoded text file.

Scoring
We'll score each submitted file individually. Your submission score is the sum of the individual submitted file scores. Your final challenge score is the sum of the best individual submitted file scores across all submissions.

If a single file doesn't comply with some problem requirement or is syntactically incorrect, you'll score no points.

We score each line of the output individually. Let's define the following line-specific measures:

- Total project cost
  For each provider and region declaration, multiply the number of packages bought for their unit cost.
  The total project cost is the sum of all these costs.
- Average latency
  For each declaration, multiply the number of packages purchased for the total sum of the number of service units composing a single package, regardless of their type.
  Add up all these numbers.
  For each declaration, multiply the number of packages bought for each service unit composing that package, and divide the number of requested resources for the obtained sum.
  For each declaration, multiply the latency between a project country and the requested provider and region for the number obtained in the point above.
  Average latency is the sum of all these numbers over all declarations of the same line.
- Availability index
  For each service type: calculate the number of units bought from all the different providers in the output file by multiplying the number of packages purchased by their number of units for that service, and take the square of that number. Take the square of each service units bought from a different provider/region for that service type, and add them all together. Divide the first number by the second.
  Take the average of this number over all the service types.
- SLA penalty
  Add up all the requested units in all declarations, regardless of the service.
  Add up all the needed units, as expressed in the input file.
  Subtract the first number from the second, and divide the result by the second number.
  Multiply the obtained factor for the base penalty to obtain the SLA Penalty.

The final line score is given by:

$$score = \frac{10^9}{\frac{[Total\ project\ cost] \times [Average\ latency]}{Availability\ index} + [SLA\ Penalty]}$$

Each individual submitted file's core is calculated by the sum of the scores of its lines.

Example

Input:

```
3 3 3 5
cpu memory disk
Italy Germany Spain
Amazon 4
Milan
60 0.32 10 5 1
50 75 52
London
100 0.8 8 8 8
75 60 35
Madrid
10 6 3 5 10
60 80 85
Moscow
10 0.1 1 10 5
50 25 70
Microsoft 2
Madrid
75 0.70 15 50 100
90 49 10
Dublin
25 1.5 12 8 24
80 45 30
Google 3
Berlin
30 1.5 40 100 500
35 10 42
Dublin
15 1 25 25 0
48 25 35
Sidney
5000 2.5 10 10 3
100 170 130
10000 Italy 1000 0 0
1000 Spain 100 60 0
255000 Italy 20 0 555
30000 Italy 250 300 780
5000000 Germany 5000 300 10000
```

Output:

Below, you'll find a possible (not necessarily the best) solution to the example input:

```
0 0 60 1 0 1 1 1 8 2 0 1 2 1 10
0 1 3 0 3 1 1 0 5
0 1 2 0 3 9 2 0 1
2 0 4 2 1 4
0 1 95 0 2 10 1 0 69 1 1 17 2 0 24 2 1 1 2 2 50
```

Scoring
Let's calculate the scores for each project:

Project	Total Project Cost	Average Latency	Availability Index	SLA Penalty	Project Score
1	60 x 0.32 + 1 x 0.7 + 8 x 1.5 + 1 x 1.5 + 10 x 1 = 43.4	(50 x 960 + 90 x 165 + 80 x 352 + 35 x 640 + 48 x 500) / (960 + 165 + 352 + 640 + 500) = 52.51	avg([(600 + 15 + 96 + 40 + 250)^2 / (600^2 + 15^2 + 96^2 + 40^2 + 250^2)]; [(300 + 50 + 64 + 100 + 250)^2 / (300^2 + 50^2 + 64^2 + 100^2 + 250^2)]; [(60 + 100 + 192 + 500)^2 / (60^2 + 100^2 + 192^2 + 500^2)]) = avg (2.31; 3.45; 2.42) = 2.73	avg(10000 x (1000 - MIN(1001;1000))/1000; 10000 x 0; 10000 x 0)) = avg (0; 0; 0) = 0	**10^9/(43.4 * 52.51/MAX(1;2.73) + 0)** **= 1196396.13**
2	3 x 0.8 + 1 x 0.1 + 5 x 0.7 = 6	(35 x 72 + 70 x 16 + 10 x 825) / (72 + 16 + 825) = 13.02	avg([(24 + 1 + 75)^2 / (24^2 + 1^2 + 75^2)]; [(24 + 10 + 250)^2 / (24^2 + 10^2 + 250^2)]; [(24 + 5 + 500)^2 / (24^2 + 5^2 + 500^2)]) = avg (1.61; 1.28; 1.12) = 1.34	avg(1000 x (100 - MIN(100;100))/100; 1000 x (60 - MIN(284;60))/60; 1000 x 0)) = avg (0; 0; 0) = 0	**10^9/(6 * 13.02/MAX(1;1.34) + 0)** **= 17088354.87**
3	2 x 0.8 + 9 x 0.1 + 1 x 1.5 = 4	(75 x 48 + 50 x 144 + 35 x 640) / (48 + 144 + 640) = 39.9	avg([(16 + 9 + 40)^2 / (16^2 + 9^2 + 40^2)]; [(16 + 90 + 100)^2 / (16^2 + 90^2 + 100^2)]; [(16 + 45 + 500)^2 / (16^2 + 45^2 + 500^2)]) = avg (2.18; 2.31; 1.25)	avg(255000 x (20 - MIN(65;20))/20; 255000 x 0; 255000 x (555 - MIN(561;555))/555)) = avg (0; 0; 0) = 0	**10^9/(4 * 39.9/MAX(1;1.91) + 0)** **= 11988281.51**

			= 1.91		
4	4 x 1.5 + 4 x 1 = 10	(35 x 2560 + 48 x 200) / (2560 + 200) = 35.94	avg([(160 + 100)^2 / (160^2 + 100^2)]; [(400 + 100)^2 / (400^2 + 100^2)]; [(2000)^2 / (2000^2)]) = avg (1.9; 1.47; 1) = 1.46	avg(30000 x (250 - MIN(260;250))/250; 30000 x (300 - MIN(500;300))/300; 30000 x (780 - MIN(2000;780))/780)) = avg (0; 0; 0) = 0	**10^9/(10 * 35.94/MAX(1;1.46) + 0)** **= 4052326.08**
5	95 x 0,8 + 10 x 6 + 69 x 0,7 + 17 x 1.5 + 24 x 1.5 + 1 x 1 + 50 x 2.5 = 371.8	(60 x 2280 + 80 x 180 + 49 x 11385 + 45 x 748 + 10 x 15360 + 25 x 50 + 170 x 1150) / (2280 + 180 + 11385 + 748 + 15360 + 50 + 1150) = 35.09	avg([(760 + 30 + 1035 + 204 + 960 + 25 + 500)^2 / (760^2 + 30^2 + 1035^2 + 204^2 + 960^2 + 25^2 + 500^2)]; [(760 + 50 + 3450 + 136 + 2400 + 25 + 500)^2 / (760^2 + 50^2 + 3450^2 + 136^2 + 2400^2 + 25^2 + 500^2)]; [(760 + 100 + 6900 + 408 + 12000 + 150)^2 / (760^2 + 100^2 + 6900^2 + 408^2 + 12000^2 + 150^2)]) = avg (4.31; 2.9; 2.15) = 3.12	avg(5000000 x (5000 - MIN(3514;5000))/5000; 5000000 x (300 - MIN(7321;300))/300; 5000000 x (10000 - MIN(20318;10000))/10000)) = avg (1486000; 0; 0) = 495333.33	**10^9/(371.8 * 35.09/MAX(1;3.12) + 495333.33)** **= 2001.93**
TOTAL					**34327360.51**

Note: the problem statement (including the customer and rates) used in Cloud Adventure is purely fictional for the purposes of the challenge.