

Grid Load

Problem Statement

Your friend Bob is the owner of a couple charging stations around the world. He made a mistake when advertising for his service, offering free charging for 100 days. He has also made a deal with the city government, which limits the amount of energy that can be drawn from the grid. Each station is tied to a car, and each car has a maximum capacity. Due to overheating issues, the stations can only charge the cars at night. Bob has analyzed the patterns of the cars and has found the different trips that each car will take. He has also found the minimum charge required to complete each trip, and the chance that this charge will be sampled. Each car will return to the same station after completing its trip. Bob wants to minimize the cost of charging the cars while ensuring that they can complete their trips. If a car cannot complete its trip, the car owner will bill Bob for using a public charging station, which has a large markup. Help Bob for 100 consecutive nights. Each night you are given updated battery levels and new prices. You reply with that night's charging plan.

The cost-function is applied per-night, but the simulation spans 100 days.

Help Bob loss the least amount of money possible.

Score Function

Let remainCharge_i be the charge for the i -th car at the start of the night. Initially

$$\text{remainCharge}_i = \frac{a_i}{2}$$

Let $\text{charge}_{i,j}$ be the charge the i -th car has at the j -th minute.

More formally,

$$\text{charge}_{i,j} = \text{remainCharge}_i + \sum_{k=0}^j s_{i,k}$$

If, at the end of the night, the charge for the i -th car is less than the required charge for a randomly sampled trip $d_{i,k}$ (where k is sampled according to $p_{i,j}$), the deficit $(d_{i,k} - \text{remainCharge}_i)^+$ must be purchased at the external price.

The penalty η for one day is:

$$\eta = \sum_{i=0}^n E_{k \sim p_{i,j}} [\max(0, d_{i,k} - \text{remainCharge}_i) \cdot \text{externalPrice}]$$

$$\mu = \sum_{i=0}^n \left[\sum_{j=0}^m (s_{i,j} \cdot \text{price}_j) \right]$$

The total cost to minimize is the total over all the days:

$$\text{Total Cost} = \mu + \eta$$

At the end of each night, remainCharge_i is updated based on the charging schedule, $s_{i,j}$ and the trip sampled. More formally, for each car i and minute j , the charge is updated as follows:

$$\text{remainCharge}_i = \text{remainCharge}_i + \sum_{j=0}^{479} (s_{i,j}) - E_{k \sim p_{i,j}}[d_{i,k}]$$

Note: the sampled trip is the same for updating remainCharge and for calculating the penalty.

Static Input Format

- The real number x : total output capacity for all stations (kW per minute).
- The integer n : number of stations (cars).
- The integer m : number of possible trips per station.
- The integer array a of length n : a_i is the capacity of the i -th car.
- The 2D array p of size $n \times m$: $p_{i,j}$ is the probability that the j -th trip is sampled for the i -th station. Each line contains m real numbers summing to 1.
- The 2D array d of size $n \times m$: $d_{i,j}$ is the minimum charge required for the j -th trip of the i -th station. Each line contains m integers.

The static input is provided in the following order:

```

x  n  m
a0 a1 ... an-1
p0,0 p0,1 ... p0,m-1
p1,0 p1,1 ... p1,m-1
⋮
pn-1,0 pn-1,1 ... pn-1,m-1
d0,0 d0,1 ... d0,m-1
d1,0 d1,1 ... d1,m-1
⋮
dn-1,0 dn-1,1 ... dn-1,m-1

```

Dynamic Input Format

- The real number array price of length 480: price_j is the price per kW at the j -th minute of the night.
- The real number externalPrice : price per kW for using an external charging station for the night.
- The real number array remainCharge of length n : remainCharge_i is the charge for the i -th car at the start of the night.

The dynamic input is provided in the following order:

price₀ price₁ ... price₄₇₉
externalPrice
remainCharge₀ remainCharge₁ ... remainCharge_{n-1}

Output Format

After each night, you must output the charging schedule for all stations. The output should be in the following format:

- Output a 2D array s of size $n \times 480$: $s_{i,j}$ is the kW allocated to the i -th station at the j -th minute.
- Each line corresponds to one station and contains 480 real numbers separated by spaces.

Output Constraints

- For every minute j ($0 \leq j < 480$): $\sum_{i=0}^{n-1} s_{i,j} \leq x$.
- For every station i and minute j : $0 \leq s_{i,j}$.
- For every station i : the total charge delivered over the night cannot exceed the car's capacity:
 $\sum_{j=0}^{479} s_{i,j} + \text{remainCharge}_i \leq a_i$.

Interaction

1. Read the static input
2. For each night:
 - The grader will provide the new dynamic input.
 - Output the charging schedule s for all stations. (*flush the output*)

Example

Input

```
5 1 2
100
0.7 0.3
3 9
```

Dynamic Input

```
0.1 0.2 0.8 ...
1000
50
```

Output

```
5.0 5.0 0.0 ...
```

Explanation

In this example, there is one station with a capacity of 100 kW. The first trip requires 3 kW with a probability of 0.7, and the second trip requires 10 kW with a probability of 0.3. The car starts with 50 kW of charge. The output schedule allocates 5 kW for the first minute, then 5 kW for the second minute, for the third minute the program decided the electricity price was too high and does not charge. The program would read the dynamic input again with the updated prices and charge of the cars, then output the schedule and repeat for 100 days.

Constraints

- $10^3 \leq x \leq 10^4$
- $1 \leq n \leq 100$
- $1 \leq m \leq 5$
- $1 \leq a_i \leq 100$
- $1 \leq d_{i,j} \leq a_i$
- $\sum_{j=0}^m (p_{i,j}) = 1$
- $0 \leq \text{price}_i \leq 1$
- $0 \leq \max_{i=0}^n (\text{price}_i) \leq \text{externalPrice} \leq 5$