Problem D. Thrill Factor

Problem Description

As Halloween approaches in a quaint, peaceful town, children eagerly prepare for a night of door-to-door trick-or-treating, infusing the town with both fear and excitement. As the orchestrator of this special event, your mission is to ensure that this night brims with spine-tingling scares.

Within the town, you'll find m children, each possessing varying levels of "trick-or-treat potential." Additionally, there are n households awaiting their share of Halloween mischief, each brimming with fear.

You have thoughtfully assembled a collection of $n \times m$ distinctive Halloween costumes for this event. Prior to the festivities, you will assign a unique number to each child, ensuring they queue up to receive their costumes in an order of your choosing. This "trick-or-treat potential" can be represented as a vector v with dimensions $m \times 1$, and you have the flexibility to arrange this order to your liking.

The event encompasses a total of n rounds of trick-or-treating. During each round, children will form a queue to obtain their Halloween costumes, creating a matrix A with dimensions $n \times m$. Here, $A_{i,j}$ signifies the "scariness" of the costume donned by the j-th child during the i-th round of trick-or-treating. The order in which the costumes are used is at your discretion. To keep the event fresh, each costume is designated for a single use, ensuring that each Halloween costume is employed exactly once and not repeated.

The "spookiness" of each round of trick-or-treating can be quantified. After the children have prepared for each round, you can compute the "spookiness" by multiplying the "trick-or-treat potential" of each child by the "scariness" of their costume and then summing these values. The result is a vector of dimensions $n \times 1$, referred to as Av, which sequentially represents the "spookiness" of each round.

Since each household has a distinct fear level when it comes to trick-or-treating, the same round of pranks may have different effects on different households. This variation can be quantified as the "thrill factor." For any given round of trick-or-treating, the "thrill factor" is determined by multiplying the household's "fear level" by the "spookiness" of that round. The order in which you visit the n households is entirely your decision, assuming that, according to your arrangement, the "fear level" of each household can be represented as a vector with dimensions $n \times 1$, denoted as u. Consequently, the overall "thrill factor" for the entire event is represented as $u^{\mathbf{T}}(Av)$.

Your goal is to obtain the maximum value of the overall "thrill factor" $u^{\mathbf{T}}Av$ for the entire event. You are to provide the "trick-or-treat potential" v in the order in which the children

should queue up, the "scariness" A of the costumes based on the order they are worn, and the "fear level" u of each household, based on your chosen order for trick-or-treating. If multiple arrangements lead to the maximum value, you have the flexibility to output any one of them.

Please keep in mind that your goal is to maximize the "thrill factor" for the entire town without arbitrarily altering the children's "trick-or-treat potential," the scariness of the prepared costumes, or the fear level of each household. Make sure that this Halloween night becomes an unforgettable experience.

Input Format

- Line 1: Two integers, n and m, representing the number of households and children, respectively.
- Line 2: A list of n integers, $u_1, u_2, \ldots, u_n$, denoting the fear levels of each household.
- Line 3: A list of m integers, $v_1, v_2, \ldots, v_m$, indicating the trick-or-treat potential of each child.
- Lines $3+i$ (where $1 \leq i \leq n$): A list of m integers, $A_{i,1}, A_{i,2}, \ldots, A_{i,m}$, representing the scariness of costumes worn by each child. Please note that you can rearrange the elements in the matrix, changing both rows and columns as needed for optimization.

Output Format

- Line 1: The maximum value of $u^{\mathbf{T}}Av$.

Subsequently, please output the modified u', v', and A' that result in the maximum value.

- Line 2: A list of n integers, $u'_1, u'_2, \ldots, u'_n$.
- Line 3: A list of m integers, $v'_1, v'_2, \ldots, v'_m$.
- Lines $3+i$ (where $1 \leq i \leq n$): A list of m integers, $A'_{i,1}, A'_{i,2}, \ldots, A'_{i,m}$, representing the scariness of costumes worn by each child during the i-th round of trick-or-treating.

Constraints

- $1 \leq n, m \leq 10$.
- $|u_i| \leq 10,000$ for $i = 1, 2, \ldots, n$.
- $|v_i| \leq 10,000$ for $i = 1, 2, \ldots, m$.
- $|A_{i,j}| \leq 10,000$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
- All inputs are integers.

Subtasks

- (20 points) You are not required to rearrange any elements in the matrix.
- (10 points) $n = 1$.
- (10 points) $u_i = 1$ for $i = 1, 2, \ldots, n$.
- (10 points) u, v, and A consists of non-negative integers.
- (50 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-5	1000	262144
1	6-11	1000	262144
2	12-15	1000	262144
3	16-20	1000	262144
4	21-25	1000	262144
5	1-29	1000	262144

Samples

Sample Input 1

```
9 8
0 -7 7 3 -4 7 -3 1 -9
-7 0 0 7 -9 -8 -5 -2
-3 -2 -1 -1 -1 -1 0 0
8 0 0 -8 9 8 7 4
-7 1 1 8 -9 -9 -7 -4
-5 1 1 5 -5 -5 -5 -3
6 2 2 -6 8 7 5 4
-7 2 2 8 -9 -8 -6 -4
5 3 3 -5 6 6 5 3
-3 3 3 3 -4 -3 -3 -3
9 3 3 -9 9 9 8 5
```

This sample input satisfies the constraints of Subtasks 1, 5.

Sample Output 1

```
11454
0 -7 7 3 -4 7 -3 1 -9
-7 0 0 7 -9 -8 -5 -2
-3 -2 -1 -1 -1 -1 0 0
8 0 0 -8 9 8 7 4
-7 1 1 8 -9 -9 -7 -4
-5 1 1 5 -5 -5 -5 -3
6 2 2 -6 8 7 5 4
-7 2 2 8 -9 -8 -6 -4
5 3 3 -5 6 6 5 3
-3 3 3 3 -4 -3 -3 -3
9 3 3 -9 9 9 8 5
```

Sample Input 2

```
1 3
1
-4 -2 -3
6 3 -4
```

This sample input satisfies the constraints of Subtasks 2, 5.

Sample Output 2

```
-5
1
-3 -4 -2
3 -4 6
```

Sample Input 3

```
3 3
1 1 1
0 -8 9
5 3 -5
2 0 -2
-2 -4 1
```

This sample input satisfies the constraints of Subtasks 3, 5.

Sample Output 3

```
178
1 1 1
0 9 -8
-2 2 -5
0 3 -4
1 5 -2
```

Sample Input 4

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

This sample input satisfies the constraints of Subtasks 4, 5.

Sample Output 4

```
169
2 1
1 4 7
2 5 6
1 3 4
```

Sample Input 5

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

This sample input satisfies the constraints of Subtask 5.

Sample Output 5

```
-345
-5 -4
-3 -6
-9 -1
-9 -3
```