NOI '12 P5 - Food Festival

Time limit: 0.6s **Memory limit:** 512M

National Olympiad in Informatics, China, 2012

To welcome students from all across the nation, City CZ is specially hosting a grand food festival.

Being a food fanatic who loves to try new flavors, little M obviously does not want to miss this feast. Going there, he quickly was able to sample lots of the festival's great foods. Yet, it is still hard to satisfy the appetite of a true gastronome, even if the dishes are very tasty and the kitchen is very efficient. Little M still thinks that not having dishes which are already on someone else's table is an unbearable feeling. So, little M started to investigate issues relating to the order in which dishes are prepared, hoping to create an efficient system that minimizes the waiting time of students.

Little M noticed that the food festival contains n varieties of delicacies. When ordering each time, each student can select one of these varieties. There are a total of m chefs to prepare these dishes. After all of the students have ordered, the tasks of preparing the dishes will be assigned to each of the chefs. Then, each chef will simultaneously start to cook. The chefs will follow the required order when they prepare the dishes, and each time can only make a portion for one person.

Other than this, little M also made an interesting observation – even though these m chefs all know how to fully prepare all n varieties of delicacies, different chefs may require different amounts of time to prepare the same dish. He numbered the types of delicacies from 1 to n, and the chefs from 1 to m, denoting the time it takes for j-th chef to prepare the i-th delicacy as $t_{i,j}$.

Little M will consider each student's *waiting time* to be from when *all chefs* start to cook to when their own dish is fully prepared. In other words, if a student's ordered dish is the k-th delicacy to be prepared by some chef, then their waiting time is the sum of the preparation times of the first k dishes that this chef prepares. The *total waiting time* is the sum of the waiting times among all of the students.

Now, little M has the orders of each and every student – there are p_i students who ordered the i-th type of delicacy (for $i = 1, 2, \dots, n$). He would like to know the minimum possible total waiting time.

Input Specification

The first line of input contains two positive integers n and m, representing the number of varieties of delicacies and chefs, respectively.
Line 2 contains n positive integers $p_1, p_2, \dots, p_n$, where p_i is the number of students who ordered the i-th delicacy.
For the following n lines, each line contains m nonnegative integers. The j-th integer on the i-th line will be $t_{i,j}$, the time it takes for the j-th chef to prepare the i-th delicacy.
All adjacent numbers on the same line will be separated by a single space, and there will be no trailing spaces on any line.

Output Specification

Output one line containing a single integer, the smallest possible total waiting time.

Sample Input

```
3 2
3 1 1
5 7
3 6
8 9
```

Sample Output

```
47
```

Explanation

Chef 1 first prepares 1 order of delicacy 2, then prepares 2 orders of delicacy 1. The waiting times of the 3 students who ordered these 3 dishes are respectively 3, $3+5=8$, and $3+5+5=13$.
Chef 2 first prepares 1 order of delicacy 1, then prepares 1 order of delicacy 3. The waiting times of the 2 students who ordered these 2 dishes are respectively 7 and $7+9=16$.
The total waiting time is $3+8+13+7+16=47$.
Although delicacies 1 and 3 are best cooked by chef 1, if these are both cooked by chef 1, then the total waiting time will actually be longer. If the solution described above is followed where 1 order of delicacy 1 and 1 order of delicacy 3 is tasked to chef 2, then chef 2 will not be idle, reducing the waiting time.
It can be proven that no solution exists which is more optimal.

Constraints

For 100% of test cases, $n \le 40, m \le 100, p \le 800$, and $t_{i,j} \le 1000$ (where $p = \sum p_i$ and the total number of ordering students).

The following additional constraints for n, m, and p will hold.

Test Case	n	m	p
1	$n=5$	$m=5$	$p=10$
2	$n=40$	$m=1$	$p=400$
3	$n=40$	$m=2$	$p=300$
4	$n=40$	$m=40$	$p=40$
5	$n=5$	$m=40$	$p=100$

Test Case	n	m	p
6	$n = 10$	$m = 50$	$p = 200$
7	$n = 20$	$m = 60$	$p = 400$
8	$n = 40$	$m = 80$	$p = 600$
9	$n = 40$	$m = 100$	$p = 800$
10	$n = 40$	$m = 100$	$p = 800$

Problem translated to English by **Alex**.