

美食节

【问题描述】

CZ 市为了欢迎全国各地的同学，特地举办了一场盛大的美食节。

作为一个喜欢尝鲜的美食客，小 M 自然不愿意错过这场盛宴。他很快就尝遍了美食节所有的美食。然而，尝鲜的欲望是难以满足的。尽管所有的菜品都很可口，厨师做菜的速度也很快，小 M 仍然觉得自己桌上没有已经摆在别人餐桌上的美食是一件无法忍受的事情。于是小 M 开始研究起了做菜顺序的问题，即安排一个做菜的顺序使得同学们的等待时间最短。

小 M 发现，美食节共有 n 种不同的菜品。每次点餐，每个同学可以选择其中的一个菜品。总共有 m 个厨师来制作这些菜品。当所有的同学点餐结束后，菜品的制作任务就会分配给每个厨师。然后每个厨师就会同时开始做菜。厨师们会按照要求的顺序进行制作，并且每次只能制作一人份。

此外，小 M 还发现了另一件有意思的事情——虽然这 m 个厨师都会制作全部的 n 种菜品，但对于同一菜品，不同厨师的制作时间未必相同。他将菜品用 $1, 2, \dots, n$ 依次编号，厨师用 $1, 2, \dots, m$ 依次编号，将第 j 个厨师制作第 i 种菜品的时间记为 $t_{i,j}$ 。

小 M 认为：每个同学的等待时间为所有厨师开始做菜起，到自己那份菜品完成为止的时间总长度。换句话说，如果一个同学点的菜是某个厨师做的第 k 道菜，则他的等待时间就是这个厨师制作前 k 道菜的时间之和。而总等待时间为所有同学的等待时间之和。

现在，小 M 找到了所有同学的点菜信息——有 p_i 个同学点了第 i 种菜品 ($i = 1, 2, \dots, n$)。他想知道的是最小的总等待时间是多少。

【输入格式】

输入文件为 *delicacy.in*。

输入文件的第 1 行包含两个正整数 n 和 m ，表示菜品的种数和厨师的数量。

第 2 行包含 n 个正整数，其中第 i 个数为 p_i ，表示点第 i 种菜品的人数。

接下来有 n 行，每行包含 m 个非负整数，这 n 行中的第 i 行的第 j 个数为 $t_{i,j}$ ，表示第 j 个厨师制作第 i 种菜品所需的时间。

输入文件中每行相邻的两个数之间均由一个空格隔开，行末均没有多余空格。

【输出格式】

输出文件为 *delicacy.out*。

输出仅一行包含一个整数，为总等待时间的最小值。

【样例输入】

3 2

3 1 1
5 7
3 6
8 9

【样例输出】

47

【样例说明】

厨师 1 先制作 1 份菜品 2，再制作 2 份菜品 1。点这 3 道菜的 3 个同学的等待时间分别为 3， $3+5=8$ ， $3+5+5=13$ 。

厨师 2 先制作 1 份菜品 1，再制作 1 份菜品 3。点这 2 道菜的 2 个同学的等待时间分别为 7， $7+9=16$ 。

总等待时间为 $3+8+13+7+16=47$ 。

虽然菜品 1 和菜品 3 由厨师 1 制作更快，如果这些菜品都由厨师 1 制作，总等待时间反而更长。如果按上述的做法，将 1 份菜品 1 和 1 份菜品 3 调整到厨师 2 制作，这样厨师 2 不会闲着，总等待时间更短。

可以证明，没有更优的点餐方案。

【数据规模及约定】

对于 100% 的数据， $n \leq 40$ ， $m \leq 100$ ， $p \leq 800$ ， $t_{i,j} \leq 1000$ （其中 $p = \sum p_i$ ，即点菜同学的总人数）。

每组数据的 n 、 m 和 p 值如下：

测试点编号	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	$n = 5$	$m = 40$	$p = 100$
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	$n = 40$	$m = 100$	$p = 800$