

任务安排(jobs.c/cpp/pas/in/out)

时限：每个测试点 2 秒

问题描述

小 Y 最近遇到了一个棘手的问题。她有两项任务需要完成，其中第一项任务是重复操作 $1(op_1)S_1$ 次，第二项任务是重复操作 $2(op_2)S_2$ 次。为了完成这些任务，小 Y 雇佣了 N 名工人。其中，第 i 个工人完成 op_1 所需时间为 $T_{1,i}$ ，完成 op_2 所需时间为 $T_{2,i}$ 。每个 op_1 和 op_2 都只能被一名工人完成，每名工人在任意时刻都只能做一项工作。

所有的工人从第 0 秒开始工作。每当一个工人开始执行一项操作(op_1 或 op_2)，他必须一直执行下去直到完成而不能被打断。我们记第一项任务完成的时间为 E_1 ，第二项任务完成的时间为 E_2 ，你的任务就是安排这些工人的工作，使得 E_1+E_2 最小。

输入数据

输入文件的第一行包含一个整数 T ，表示输入文件中数据的组数。

每个测试数据的第一行包含三个整数 $N S_1 S_2$ ，含义如上文所述。

接下来的 N 行每行包含两个整数 $T_{1,i}$ 、 $T_{2,i}$ ，分别表示第 i 个工人完成 op_1 和 op_2 所需的时间。

输出数据

输出文件包含 T 行，每行只有一个整数，表示你找到的 E_1+E_2 的最小值。

输入样例

```
4

1 2 3
10 20

3 5 7
10 20
15 16
17 18

4 3 6
10 12
8 9
16 11
13 20

4 4 6
7 12
5 3
6 5
1000000 1000000
```

输出样例

100

162

84

41

样例说明

第一组数据中，唯一的工人首先执行 2 次 op_1 ，在第 20 秒完成任务一($E_1=20$)。然后执行 2 次 op_2 ，在第 80 秒完成任务二($E_2=80$)。因此答案为 $20+80=100$ 。

第二组数据中，工人#1 连续执行 5 次 op_1 ，在第 50 秒完成任务一($E_1=50$)，工人#2 执行 7 次 op_2 ，在第 112 秒完成任务二($E_2=112$)。因此答案为 $50+112=162$ 。

第三组数据和第二组数据类似。

第四组数据中，工人#2 首先连续执行 6 次 op_2 ，在第 18 秒完成任务二($E_2=18$)。于此同时，工人#3 执行 3 次 op_1 ，同样在第 18 秒完成。此时还需要执行一次 op_1 ，因此让工人#2 去执行最后一次 op_1 ，在第 23 秒完成任务一($E_1=23$)、因此答案为 $18+23=41$ 。

数据规模

100%的数据中， $1 \leq T \leq 7, 1 \leq N \leq 100, 1 \leq S_1, S_2 \leq 7, 1 \leq T_{1,i}, T_{2,i} \leq 1000000$ 。