## Exercise – *Canteen*

At AlgoUniversity, the *canteen* is operated by an external caterer. AlgoUni demands quite a lot from the canteen chef: the menu price is fixed by the university for each day (but prices may differ for different days). Furthermore every student who wants to eat at the canteen on a given day should be served whenever possible. In order to help the canteen to plan accordingly, the students have preregistered.

Due to the availability of workers and food, the canteen can only produce a limited amount of menus per day, and menu production costs can vary as well. To save costs, the chef has a brilliant (though not so tasty) idea: He might produce more menus than needed and store the leftovers in the freezer to serve them on any later day. If it is not possible to serve all the students the chef can decide who gets served and who does have to stay hungry as long as he serves as many students as possible.

Since this makes planning quite difficult, you are hired by the chef to figure out whether it's possible to satisfy AlgoUni's demands at all and how much revenue (or loss) can be achieved by the canteen.

**Input** The first line of the input contains the number $t \leqslant 30$ of test cases. Each of the $t$ test cases is described as follows.

- It starts with a line containing one integer $n$, $1 \leqslant n \leqslant 1000$, denoting the number of days for which the chef wants to plan ahead. The following prescribes the conditions for the canteen:
- The second line contains $2n$ space seperated integers $a_1, c_1, a_2, c_2, \ldots, a_n, c_n$ denoting for each day $i$ the amount of menus $0 \leqslant a_i \leqslant 300$ that can be produced and the production cost per menu $0 \leqslant c_i \leqslant 20$.
- The third line contains $2n$ space seperated integers $s_1, p_1, s_2, p_2, \ldots, s_n, p_n$ denoting for each day $i$ the number of students $0 \leqslant s_i \leqslant 300$ who want to eat at the canteen and the menu price $0 \leqslant p_i \leqslant 20$ set by the university.
- The last line defines the freezer volume and energy cost for each night. These are given by $2n-2$ space seperated integers $v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}$ which denote the number of menus $0 \leqslant v_i \leqslant 300$ which can be stored overnight from day $i$ to day $i+1$ as well as the cooling energy cost per menu $0 \leqslant e_i \leqslant 20$.

**Output** For each test case output a line containing the word `possible`, if the canteen can serve all the students of AlgoUniversity and `impossible` otherwise. For the rest of the line, print two integers S and P, the maximum number of students that can be served and the maximum profit (or minimum loss) the canteen can achieve.

**Points** There are three groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 30 points, you may assume that the menu production costs and the menu prices are fixed over time, and that freezing menus costs nothing. (Formally $\forall i,\ 1 \leqslant i \leqslant n-1\colon c_{i+1} = c_i,\ p_{i+1} = p_i,\ e_i = 0$.)
2. For the second group of test sets, worth 50 points, you may assume that $n \leqslant 250$.
3. For the third group of test sets, worth 20 points, there are no additional assumptions.

Corresponding sample test sets are contained in `testi.in/out`, for $i \in \{1, 2, 3\}$.