UCF Local Contest (Round 1B) — September 12, 2020

Make the Team

filename: `maketeam`
Difficulty Level: `Medium-Hard`
Time Limit: `5 seconds`

You are eager to make the programming team. You have decided that if you watch several videos, you will increase your chances of making the team. Each video you want to watch is one hour long, and each video plays at specific times. Naturally, you want to get through all the videos as fast as possible, to leave more time to practice!

For example, if you want to watch two videos and the first one is available at the time intervals [1,2), [4,5), [8,9) and [12,13), and the second video is available at the time intervals [4,5), [7,8), and [11,12), then the earliest time at which you can complete the two videos is time 5. You can accomplish this by watching the first video in the time interval [1,2) and the second one at the time interval [4,5). Note that all videos play for precisely the length of one time interval, and one can watch back to back videos. Thus, if one video plays at the interval [x, x+1) and another video plays at the interval [x+1, x+2), where x is a positive integer, both can be watched, back to back.

The Problem:

Given a list of times that each video you want to watch is available, determine the earliest time at which you can complete watching all of the videos. Note that the videos can be watched in any order as long as the time intervals allow.

The Input:

The first input line contains a single integer, n ($1 \le n \le 200$), indicating the number of videos you would like to watch. Each of the next n input lines describes a video you want to watch. The i^{th} of these input lines starts with an integer, t_i ($1 \le t_i \le 30$), representing the number of times video i is available to watch. This is followed by t_i space separated values indicating the starting time video i is available to watch. The list of times for each video will be a strictly increasing list of positive integers, with a maximum value of 1000. It is guaranteed that there will be at least one arrangement that allows you to watch all of the videos.

The Output:

Print the earliest time at which you can complete watching all of the videos.

Sample Input	**Sample Output**

3
2 4 6
3 4 9 11
1 4
	10

4
2 3 11
3 2 9 11
2 2 3
2 3 9
	12