

Problem H: Hackathon

Every year the **Bright Developers Initiative (BDI)** runs its **Youth Instigators Hackathon** at the KKU Science Park, and this year's theme is "**Living Smarter.**" The event lasts several days, a rating of each team is evaluated at the end of each day, and whoever has the highest rating at the end of the final day lifts the championship trophy.

Your team should not even be nervous. Your teammates are quietly brilliant, and you can steer your own daily rating to whatever value you like. Winning outright would be easy, and that is exactly the problem: it is boring. You have watched far too many underdog documentaries on Hackflix, and you want the story. You want to look hopeless for the first half of the event, hiding at the very bottom of the board, and then climb out of nowhere to snatch first place on the very last day.

There is one catch. To keep the comeback looking natural rather than suspicious, your rating must rise by the **same whole number every single day**. No dramatic jumps, no clever dips, just a steady honest climb that happens to end on top.

The hackathon runs for **N** days. There are **M** other teams, and you are given the rating of every other team at the end of each day; team ratings are fixed and known in advance.

You can manage your rating at will: you make your rating equals to a starting value **s** at the end of first day, and the value will increase by a value **k** daily, where both **s** and **k** are integers with **$s \geq 1$** and **$k \geq 1$** (so your rating strictly increases by the same amount each day).

You want to choose **s** and **k** so that:

1. **You look like an underdog.** On every day of the first half of the event (days 1 through $\lfloor N/2 \rfloor$), your rating is strictly lower than the rating of **every team** on that day.
2. **You win at the end.** On day **N** , your rating is strictly higher than the rating of **every team**.

Among all plans that pull this off, you want the most dramatic finish possible: the **smallest** final-day rating that still makes you champion.

Input:

The first line contains an integer T , the number of test cases.

For each test case:

- a line with two integers N and M : the number of days and the number of other teams;
- then M lines, each with N integers, where the d -th integer on a line is that team's rating at the end of day d .

Output:

For each hackathon, print on its own line the minimum final-day rating your team needs, or -1 if the underdog plan is impossible.

Sample Input	Sample Output
3 4 1 3 5 6 9 3 3 10 20 50 1 40 60 30 35 40 6 2 1000 1100 1200 1300 1400 2025 1500 1500 1500 1500 1500 1900	10 -1 2026

Explanation:

- **Test Case #1:** there are four days, so the first half is days 1 and 2. Choosing $s = 1$ and $k = 3$, your rating at the end of each day is 1, 4, 7, 10. This makes your rating lower than the other team in the first two days and becomes higher at the end.
- **Test Case #2:** this is not possible because one of team has rating of 1 on day 1, so we cannot be possibly be lower than 1 (s must be positive)

Constraints:

- $1 \leq T \leq 100$
- $2 \leq N \leq 2,000$
- $1 \leq M \leq 50$
- $1 \leq \text{every given team rating} \leq 100,000$