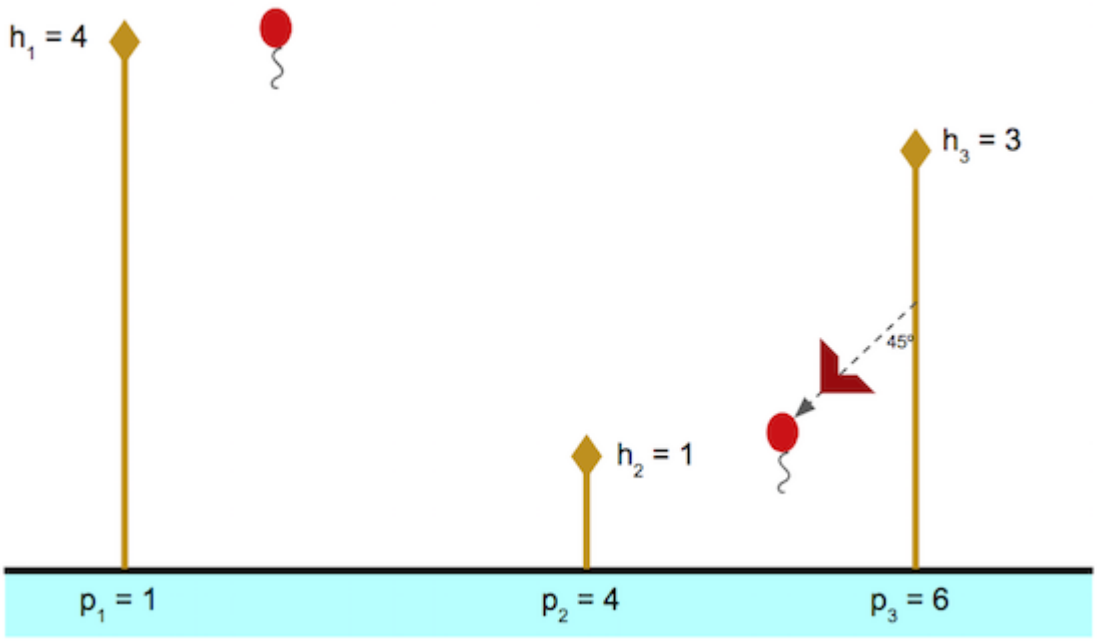
Kick Start 2018 - Round D

Paragliding

Problem



In order to advance in his quest to defeat the villainous Graphendorf, our hero Edge has to overcome a challenge in a shrine. The shrine is two-dimensional in the xy-plane, and there are **N** towers (of varying heights) erected along the x-axis of the plane. Each tower can be represented by a vertical line segment in a 2D plane. For tower i, the base of the tower is at ($\mathbf{p}_i$, 0), and the top of the tower is at ($\mathbf{p}_i$, $\mathbf{h}_i$). There are also **K** balloons floating in the plane. Each balloon can be represented by a single point in the 2D plane, with balloon i at position ($\mathbf{x}_i$, $\mathbf{y}_i$). Edge has to collect as many balloons as possible in this challenge.

Fortunately, Edge has a trusty paraglider which he found in a different room of this shrine. He may choose to climb any tower and glide down from any position on the tower towards either the positive or negative direction of the x-axis. When he glides, he descends in a straight path that makes a 45 degrees angle relative to the tower. Edge can collect any balloons in his way when he glides down from the tower. He can repeat this process of climbing up a tower and jumping off any number of times. If he touches a tower during his descent, then he is considered to be on the tower at the point and climbing. You may assume Edge to be a single point in the xy-plane.

Using a pair of goggles made from ancient technology, Edge was able to figure out the height and position of each tower and balloon. With this information, can you help Edge deduce the maximum number of balloons that he can collect in this shrine?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case contains five lines. The first line contains the integers **N** and **K** as described above. Each of the next four lines describe a recurrence used to generate the positions and heights of the towers and the x and y coordinates of the balloons. The four lines will each contain six integers in the following format:

- $\mathbf{p}_1$ $\mathbf{p}_2$ $\mathbf{A}_1$ $\mathbf{B}_1$ $\mathbf{C}_1$ $\mathbf{M}_1$
- $\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{A}_2$ $\mathbf{B}_2$ $\mathbf{C}_2$ $\mathbf{M}_2$
- $\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{A}_3$ $\mathbf{B}_3$ $\mathbf{C}_3$ $\mathbf{M}_3$
- $\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{A}_4$ $\mathbf{B}_4$ $\mathbf{C}_4$ $\mathbf{M}_4$

To generate the values for $\mathbf{p}_i$ (from 3 to **N**), $\mathbf{h}_i$ (from 3 to **N**), $\mathbf{x}_i$ (from 3 to **K**) and $\mathbf{y}_i$ (from 3 to **K**), we use the following recurrences:

- $\mathbf{p}_i = (\mathbf{A}_1 \times \mathbf{p}_{i-1} + \mathbf{B}_1 \times \mathbf{p}_{i-2} + \mathbf{C}_1)$ modulo $\mathbf{M}_1 + 1$, for i = 3 to **N**.
- $\mathbf{h}_i = (\mathbf{A}_2 \times \mathbf{h}_{i-1} + \mathbf{B}_2 \times \mathbf{h}_{i-2} + \mathbf{C}_2)$ modulo $\mathbf{M}_2 + 1$, for i = 3 to **N**.
- $\mathbf{x}_i = (\mathbf{A}_3 \times \mathbf{x}_{i-1} + \mathbf{B}_3 \times \mathbf{x}_{i-2} + \mathbf{C}_3)$ modulo $\mathbf{M}_3 + 1$, for i = 3 to **K**.
- $\mathbf{y}_i = (\mathbf{A}_4 \times \mathbf{y}_{i-1} + \mathbf{B}_4 \times \mathbf{y}_{i-2} + \mathbf{C}_4)$ modulo $\mathbf{M}_4 + 1$, for i = 3 to **K**.

It is guaranteed that no two towers share the same position. However, it is possible for a tower to overlap with a balloon. In this case, we assume that the balloon can be collected. Note that two or more balloons might share a point; in that case, Edge can collect all of those balloons at once by passing through that point.

Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the maximum number of balloons that Edge can collect.

Limits

$1 \le \mathbf{T} \le 100$.
Time limit: 40 seconds per test set.
Memory limit: 1 GB.
$0 \le \mathbf{A}_i \le 10^9$ for i = 1 to 4.
$0 \le \mathbf{B}_i \le 10^9$ for i = 1 to 4.
$0 \le \mathbf{C}_i \le 10^9$ for i = 1 to 4.
$1 \le \mathbf{M}_i \le 10^9$ for i = 1 to 4.
$1 \le \mathbf{p}_1, \mathbf{p}_2 \le 10^9$.
$1 \le \mathbf{h}_1, \mathbf{h}_2 \le 10^9$.
$1 \le \mathbf{x}_1, \mathbf{x}_2 \le 10^9$.
$1 \le \mathbf{y}_1, \mathbf{y}_2 \le 10^9$.
$p_i \neq p_j$ for i, j = 1 to **N**, i $ne; j.

Small dataset (Test set 1- Visible)

$2 \le \mathbf{N} \le 1000$.
$2 \le \mathbf{K} \le 1000$.

Large dataset (Test set 2 - Hidden)

$2 \le \mathbf{N} \le 10^5$.
$2 \le \mathbf{K} \le 10^5$.

Sample

Sample Input

```
2
3 2
1 4 1 1 0 11
4 1 1 1 8 11
2 5 0 0 0 11
4 1 0 0 0 11
5 5
2 4 1 0 1 13
4 4 0 1 12 13
1 4 1 1 0 13
3 5 1 1 7 13
```

Sample Output

```
Case #1: 1
Case #2: 4
```

Note that the input for Sample Case #1 produces the scenario depicted in the problem statement. The generated arrays are:

- **p** = [1, 4, 6].
- **h** = [4, 1, 3].
- **x** = [2, 5].
- **y** = [4, 1].

In Sample Case #2, the generated arrays are:

- **p** = [2, 4, 6, 8, 10].
- **h** = [4, 4, 4, 4, 4].
- **x** = [1, 4, 6, 11, 5].
- **y** = [3, 5, 3, 3, 1].