Kick Start 2018 - Round D

Candies

Problem

Supervin loves to eat candies. Today, his favorite candy shop is offering **N** candies, which are arranged in a line. The i-th candy in the line (counting starting from 1) has a sweetness level $\mathbf{S}_i$. Note that the sweetness level of a candy might be negative, which means the candy tastes bitter.

Supervin likes to eat sweet candies. However, candies with a combined sweetness level of more than **D** would be too much sweetness even for him. Supervin also realises that a candy with an odd sweetness level is "odd", and he does not want to eat more than **O** odd candies. In other words, an odd candy is a candy with a sweetness level that is not evenly divisible by 2. Additionally, since Supervin is in a rush, he can only eat a single contiguous subset of candies.

Therefore, he wants to eat a contiguous non-empty subset of candies in which there are at most **O** odd candies and the total sweetness level is maximized, but not more than **D**. Help Supervin to determine the maximum total sweetness level he can get, or return `IMPOSSIBLE` if there is no contiguous subset satisfying these constraints.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case contains two lines. The first line contains three integers **N**, **O**, and **D**, as described above. The second line contains seven integers $\mathbf{X}_1$, $\mathbf{X}_2$, **A**, **B**, **C**, **M**, **L**; these values are used to generate the values $\mathbf{S_i}$, as follows:

We define:

- $\mathbf{X}_i = (\mathbf{A} \times \mathbf{X}_{i-1} + \mathbf{B} \times \mathbf{X}_{i-2} + \mathbf{C})$ modulo **M**, for i = 3 to **N**.
- $\mathbf{S}_i = \mathbf{X}_i + \mathbf{L}$, for i = 1 to **N**.

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 total sweetness level Supervin can get, or `IMPOSSIBLE` if there is no possible contiguous subset satisfying the problem constraints.

Limits

$1 \le \mathbf{T} \le 100$.
Time limit: 40 seconds per test set.
Memory limit: 1 GB.
$2 \le \mathbf{N} \le 5 \times 10^5$.
$0 \le \mathbf{O} \le \mathbf{N}$.
$-10^{15} \le \mathbf{D} \le 10^{15}$.
$0 \le \mathbf{X}_1, \mathbf{X}_2, \mathbf{A}, \mathbf{B}, \mathbf{C} \le 10^9$.
$1 \le \mathbf{M} \le 10^9$.

Small dataset (Test set 1 - Visible)

L = 0.

Large dataset (Test set 2 - Hidden)

$-5 \times 10^8 \leq \mathbf{L} \leq 0$.

Sample

Note: there are additional samples that are not run on submissions down below.

Sample Input

```
2
6 1 1000000000000000
1 1 1 1 0 100 0
6 1 -100
1 1 1 1 0 100 0
```

Sample Output

```
Case #1: 13
Case #2: IMPOSSIBLE
```

In Sample Case #1, the generated array of sweetness values $\mathbf{S_i}$ is: [**1**, **1**, 2, **3**, **5**, 8], where the bold and underlined numbers are the odd numbers. Since Supervin can only eat one odd candy, he can get a maximum total sweetness level by taking the fifth and the sixth candies.

In Sample Case #2, the generated array of sweetness values $\mathbf{S_i}$ is the same as in Sample Case #1. However, this time Supervin cannot eat candies with a total sweetness level of more than -100, so no contiguous subset of candies satisfies the constraints.

Note: We do not recommend using interpreted/slower languages for the Large dataset of this problem.

Additional Sample - Test Set 2

The following additional sample fits the limits of Test Set 2. It will not be run against your submitted solutions.

Sample Input

```
3
10 1 8
4 3 4 1 5 20 -10
10 2 8
4 3 4 1 5 20 -10
10 1 8
4 3 4 1 5 20 -19
```

Sample Output

```
Case #1: 7
Case #2: 8
Case #3: -5
```

In Sample Case #1, the generated array of sweetness values $\mathbf{S_i}$ is: [-6, **-7**, **-9**, 2, 4, **3**, **1**, -8, -6, **-7**], where the bold and underlined numbers are the odd numbers. Since Supervin can only eat one odd candy and he cannot eat candies with a total sweetness level of more than 8, he can get the maximum total sweetness level by taking the fifth and the sixth candies.

In Sample Case #2, the generated array of sweetness values $\mathbf{S_i}$ is the same as in Sample Case #1. However, this time Supervin can eat two odd candies. Therefore, he can get a maximum total sweetness level by taking the fifth, the sixth, and the seventh candies.

In Sample Case #3, the generated array of sweetness values $\mathbf{S_i}$ is: [**-15**, -16, -18, **-7**, **-5**, -6, -8, **-17**, **-15**, -16] where the bold and underlined numbers are the odd numbers. Note that it is possible for the maximum total sweetness level to be negative.