This is an unofficial file made by me with the purpose of compiling all problems into a single file.

Re-sorting the Array with Scores

We define one modification for integer n as:

If n is even, let $n = n/2$;

If n is odd, let $n = 3n + 1$.

We define the number of consecutive modifications to transform n to 1 as the score of n. For example, the score of 21 is 7 ($21 \to 64 \to 32 \to 16 \to 8 \to 4 \to 2 \to 1$).

In this question, given 3 integers x, y, i, you are required to re-sort all the integers inside $[x, y]$ in the ascending order of scores, and then report the i-th number. Notice that if 2 integers have the same score, sort them in the ascending order of its absolute value.

Sample Input

A string split by space (' '), including x, y, i from left to right.

```
1 4 3
```

Sample Output

An integer.

```
4
```

Explanation of Sample

The score of 1 is 0, the score of 2 is 1, the score of 3 is 7, the score of 4 is 2. The re-sorted array is: 1, 2, 4, 3, and the 3rd number is 4.

Constraints of 10 test cases

- Case 1: $1 \leq x \leq y \leq 10$
- Case 2-4: $10 \leq x \leq y \leq 100$
- Case 5-7: $100 \leq x \leq y \leq 1000$
- Case 8-10: $1000 \leq x \leq y \leq 10000$
- $1 \leq i \leq y - x + 1$

Gas Station Planning

In somewhere sparsely populated, there is one single newly-built highway connecting multiple villages. To save energy, you are required to arrange n gas stations on this highway, minimizing the sum of distances between every village and its nearest gas station.

We can treat this highway as a one-dimension space. In this question, given an array of integers x and an integer n, where x collects the coordinates of all the villages and n is the desired number of gas stations, you are required to output the minimal sum of distances between every village and its nearest gas station.

Hint

1. In this question, scenarios where n larger than length(x) are trivial. (Why?) You need to be cautious when considering these cases.
2. How to arrange the gas station when $n = 1$?

Sample Input

The first line (x) is a string split by space (' '), including the coordinates of villages.
The second line (n) is a string, indicating the number of gas stations.

```
2 3 17 10 8
3
```

Sample Output

An integer. (It cannot be fractional. Why?)

```
3
```

Explanation of Sample

The 3 gas stations should be arranged between 2 and 3, between 8 and 10, and at 17. The sum of distances is $(3-2)+(10-8)+0=3$.

Constraints of 10 test cases

- $1 \leq x[i] \leq 10000$
- Case 1: $1 \leq \text{length}(x) \leq 10$
- Case 2-5: $10 \leq \text{length}(x) \leq 100$
- Case 6-10: $100 \leq \text{length}(x) \leq 1000$
- $1 \leq n \leq \text{length}(x)$

Number of Rigorous Subsequences

We claim a sequence Z is **rigorous** if it consists of one or more '2's, followed by one or more '1's, and then one or more '0's. For example, $[2,2,2,1,0,0]$ and $[2,1,0]$ are both rigorous, while $[1,2]$, $[1,2,0]$, and $[1,1,0]$ are not.

In this question, given an integer sequence X including only 0, 1, 2, you are required to output the number of different **rigorous** subsequences of X. Because the result might be very large, you need to print the output modulo 10^8+7. Please calculate the remainders frequently to avoid integer overflow, especially when you are using Java and C++.

A sequence Z is a **subsequence** of X if Z can be obtained from X by deleting symbols. Notice that two subsequences are different as long as they are generated by deleting symbols at different positions, despite they might look the same (Please check the sample input and output).

Sample Input

A string split by space (' '), including the symbols in X from left to right.

```
2 1 1 0
```

Sample Output

An integer.

```
3
```

Explanation of Sample

All the 3 rigorous subsequences are in bold: $[\mathbf{2},\mathbf{1},\mathbf{1},\mathbf{0}]$, $[\mathbf{2},\mathbf{1},1,\mathbf{0}]$, $[\mathbf{2},1,\mathbf{1},\mathbf{0}]$

Constraints of 10 test cases

- $X[i] \in \{0,1,2\}$
- Case 1: $1 \leq X.length \leq 100$
- Case 2-4: $100 \leq X.length \leq 1000$
- Case 5-7: $1000 \leq X.length \leq 10000$
- Case 8-10: $10000 \leq X.length \leq 100000$

Save Cost for Cargo Port

You are the manager of a cargo port, where multiple cargo ships come to carry goods. Specifically, the i-th ship has to finish n_i days of loading work during a certain period (from the x_i-th day to the y_i-th day), where the loading days are not necessarily continuous.

Now a loading plan for L ships (represented by a list M) is provided, where $M[i] = [x_i, y_i, n_i]$. To save operating cost, you should arrange the ships to load as simultaneously as possible, and close the port on days when no ship comes to load. In this question, L and M are given, and you are required to output the minimal number of days when you have to open the port.

Sample Input

The first line (L) is a string, indicating the number of ships in the loading plan (the length of M).

The remaining L lines represent loading arrangements for all the ships. In each line, x_i, y_i, n_i are included from left to right.

```
3
3 4 1
7 8 1
2 8 2
```

Sample Output

An integer.

```
2
```

Explanation of Sample

The input displays a loading plan for 3 ships. The first ship has to finish the loading work from day 3 to day 4, and it takes 1 day to finish the loading. The second ship has to finish the loading work from day 7 to day 8, and it takes 1 day to finish the loading. The third ship has to finish the loading work from day 2 to day 8, and it takes 2 days to finish the loading. In this example, the best arrangement is to let the first ship load on day 4, the second ship on day 7, and the third ship on days 4 and 7 (the loading days can be discontinuous). In this case, you only need to open the port on days 4 and 7, and the output should be 2.

Constraints of 10 test cases

- $1 \leq n_i \leq y_i - x_i + 1$
- Case 1: $1 \leq L \leq 10, 1 \leq x_i \leq y_i \leq 10$
- Case 2-4: $10 \leq L \leq 100, 10 \leq x_i \leq y_i \leq 100$
- Case 5-7: $100 \leq L \leq 1000, 100 \leq x_i \leq y_i \leq 1000$
- Case 8-10: $1000 \leq L \leq 10000, 1000 \leq x_i \leq y_i \leq 10000$