Sample Input and Output with Explanation

Input Format:

1. The first line contains two integers `n` (number of plants) and `m` (number of zombies).
2. The second line contains `n` integers representing the attack power of the plants.
3. The third line contains `n` integers representing the attack delay of the plants (in seconds).
4. The fourth line contains `m` integers representing the initial health of the zombies.
5. The fifth line contains `m` integers representing the walking speed of the zombies (in units/second).
6. The sixth line contains an integer `initialDistance` (same for all zombies, in units).
7. The seventh line contains `m` integers, where `targets[j]` specifies the index of the plant that attacks zombie `j`.

Output Format:

1. An array representing the **final health** of each zombie after the battle.
2. An integer representing the **total number of zombies defeated**.

Example Input:

```
2 3

50 30

2 1

100 150 200

2 3 4

10

0 1 1
```

Example Output:

```
0 60 140

2
```

Explanation:

P0											**Z0**
P1											**Z1, Z2**

P0: Attack = 50, Delay = 2 sec
P1: Attack = 30, Delay = 1 sec

Z0: Health = 100, Speed = 2 sec/unit
Z1: Health = 150, Speed = 3 sec/unit
Z2: Health = 200, Speed = 4 sec/unit

1. **Zombie 0:**
   - Time to reach: initialDistance / speed[0] = 10 / 2 = 5 seconds
   - Maximum attacks: floor(5 / attackDelay[0]) = floor(5 / 2) = 2
   - Plant 0 attacks twice: Health = 100 → 50 → 0. Zombie 0 is defeated before reaching the defense line.
2. **Zombie 1:**
   - Time to reach: initialDistance / speed[1] = 10 / 3 = 3 seconds
   - Maximum attacks: floor(3 / attackDelay[1]) = floor(3 / 1) = 3
   - Plant 1 attacks three times: Health = 150 → 120 → 90 → 60. Zombie 1 successfully crosses the defense line with a final health of 60.
3. **Zombie 2:**
   - Time to reach: initialDistance / speed[2] = 10 / 4 = 2.5 seconds
   - Maximum attacks: floor(2.5 / attackDelay[1]) = floor(2.5 / 1) = 2
   - Plant 1 attacks twice: Health = 200 → 170 → 140. Zombie 2 successfully crosses the defense line with a final health of 140.

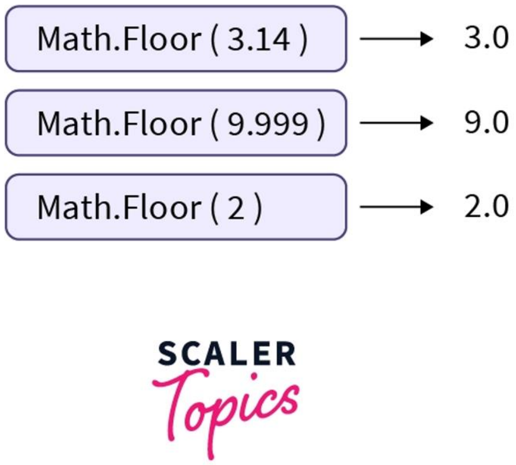
Additional Resource: Math.floor()

The `floor()` method rounds a number DOWN to the nearest integer [1].

Syntax: `Math.floor(double value)`

A double value is passed to the `Math.floor()` function.

For Example [2]:



References:

[1] "Java Math floor() Method", *w3schools*. https://www.w3schools.com/java/ref_math_floor.asp

[2] A. Kumar, "Math floor() Java | Math.floor() Function in Java - Scaler Topics," *Scaler Topics*, Mar. 30, 2022. https://www.scaler.com/topics/math-floor-java/ (Accessed Nov. 26, 2024).