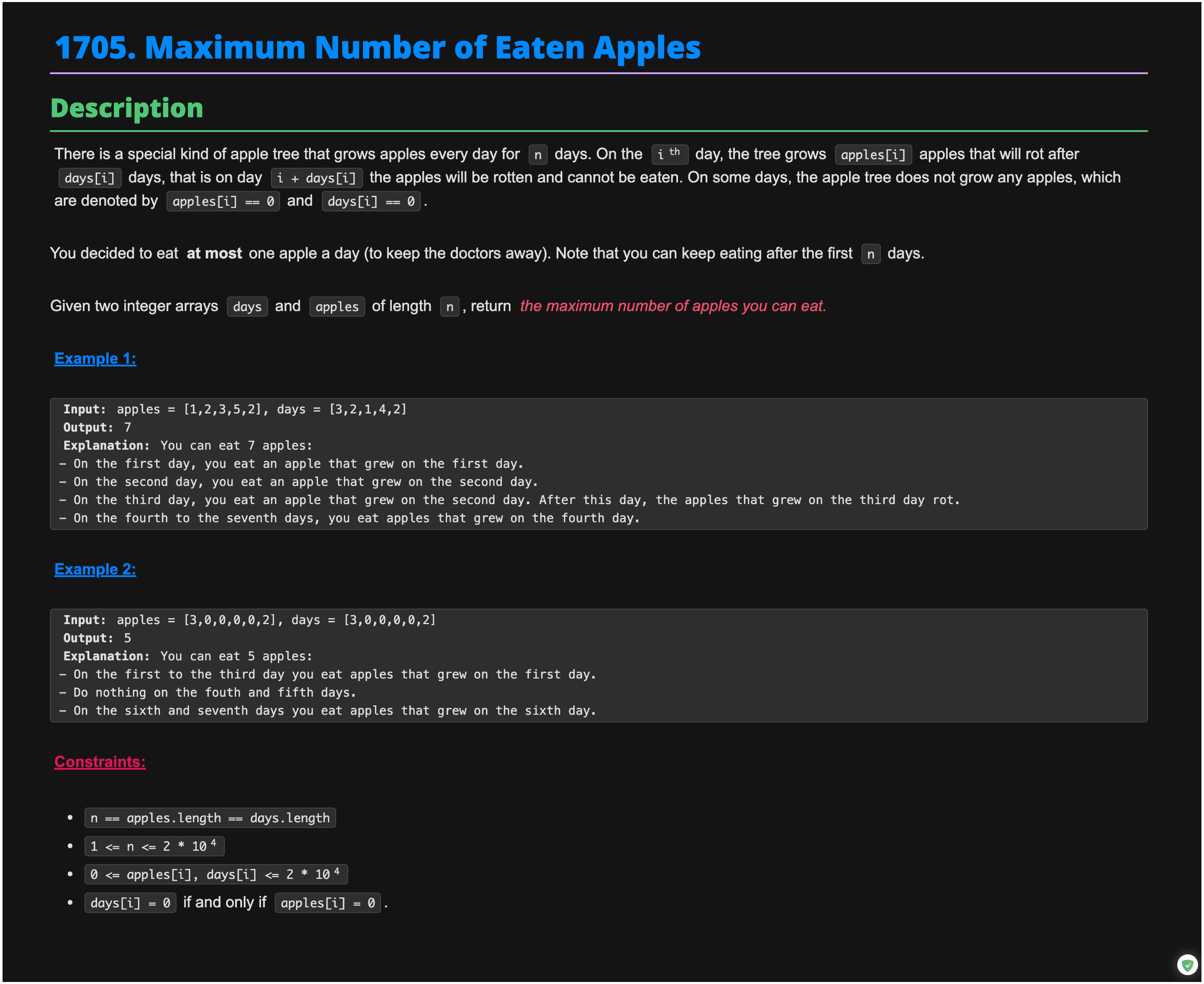

1705. Maximum Number of Eaten Apples

Description

There is a special kind of apple tree that grows apples every day for `n` days. On the `i`th day, the tree grows `apples[i]` apples that will rot after `days[i]` days, that is on day `i + days[i]` the apples will be rotten and cannot be eaten. On some days, the apple tree does not grow any apples, which are denoted by `apples[i] == 0` and `days[i] == 0`.

You decided to eat **at most** one apple a day (to keep the doctors away). Note that you can keep eating after the first `n` days.

Given two integer arrays `days` and `apples` of length `n`, return *the maximum number of apples you can eat.*

Example 1:

```
Input: apples = [1,2,3,5,2], days = [3,2,1,4,2]
Output: 7
Explanation: You can eat 7 apples:
- On the first day, you eat an apple that grew on the first day.
- On the second day, you eat an apple that grew on the second day.
- On the third day, you eat an apple that grew on the second day. After this day, the apples that grew on the third day rot.
- On the fourth to the seventh days, you eat apples that grew on the fourth day.
```

Example 2:

```
Input: apples = [3,0,0,0,0,2], days = [3,0,0,0,0,2]
Output: 5
Explanation: You can eat 5 apples:
- On the first to the third day you eat apples that grew on the first day.
- Do nothing on the fouth and fifth days.
- On the sixth and seventh days you eat apples that grew on the sixth day.
```

Constraints:

- `n == apples.length == days.length`
- `1 <= n <= 2 * 10`^{4}
- `0 <= apples[i], days[i] <= 2 * 10`^{4}
- `days[i] = 0` if and only if `apples[i] = 0`.