Offline Coding Challenges

2 exercises
Please write your solution in one of the following languages: Java, C#, Python, or Perl

1. Road Work Project

A construction company is paving a road and the township wants to know when the work will be done. The road has **X** segments to be paved.

A is a zero-indexed integer array of length **N** that provides work estimates for the project.

The index of **A** represents the months since the project began.
The values of **A** represent the segment of the road that can be completed that month.
The project will be completed when all the segments up to and including **X** are complete.

The goal is to find the earliest month that the project can be completed.

For example, you are given integer **X = 5** and array **A** such that:

```
A[0] = 5 // in month 0 the contractor can pave segment 5
A[1] = 3 // in month 1 the contractor can pave segment 3
A[2] = 4 // in month 2 the contractor can pave segment 4
A[3] = 4 // in month 3 the contractor can pave segment 4
A[4] = 2 // in month 4 the contractor can pave segment 2
A[5] = 3 // in month 5 the contractor can pave segment 3
A[6] = 1 // in month 6 the contractor can pave segment 1
A[7] = 4 // in month 7 the contractor can pave segment 4
```

In this example, the correct answer to the earliest month that the contractor can finish paving all **5** segments of the road is month 6

Given a non-empty zero-indexed array **A** consisting of **N** integers and integer **X**, write a function in your preferred language that returns the earliest time the contractor can finish paving the road.

If the contractor will never be able to completely pave the road, the function should return −1.

For example, given **X** = 5 and array **A** such that:

```
A[0] = 5
A[1] = 3
A[2] = 4
A[3] = 4
A[4] = 2
A[5] = 3
A[6] = 1
A[7] = 4
```

the function should return 6, as explained above.

Assume that:

> **N** and **X** are integers within the range [1..100,000];
> each element of array **A** is an integer within the range [1..**X**].

Complexity:

> expected worst-case time complexity is: O(N)
> expected worst-case space complexity is:
> (storage required for reading in arguments) + O(X)

Elements of the input array can be modified.

2. Crossing Paths

A non-empty zero-indexed array **A** consisting of **N** integers is given. The elements of array **A** represent hikers traveling either up or down a trail.

Array A contains only 0s and/or 1s:

- 0 represents a hiker going up the trail.
- 1 represents a hiker going down the trail.

We want to know how many times a pair of hikers will cross paths on the trail.

For example, consider array A such that:

```
A[0] = 0
A[1] = 1
A[2] = 0
A[3] = 1
A[4] = 1
```

We have five pairs of hikers crossing paths: (0, 1), (0, 3), (0, 4), (2, 3), (2, 4).

Write a function called *solution* in your preferred language that takes the integer array A as an argument and returns the total number of crossing pairs as an int.

The function should return −1 if the number of pairs of crossing exceeds 1,000,000,000.

Assume that:

- **N** is an integer within the range [1..100,000];
- Each element of array **A** is an integer that has the values 0 or 1.

Complexity:

expected worst-case time complexity is O(N);

expected worst-case space complexity is:

(storage required for reading in arguments) + O(1)

Elements of input arrays can be modified.