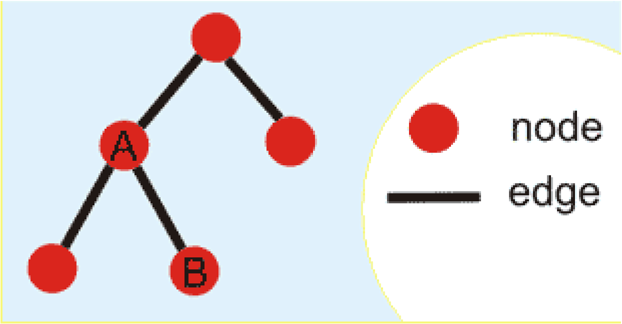
Problem Statement (P8XGraphBuilder)

You want to build a graph consisting of N nodes and N-1 edges. The graph must be connected.



The degree of a node in the graph is equal to the number of edges adjacent to the node. For example, the degree of node A in the picture above is 3, while the degree of node B is 1. Note that in your graph the degree of each node will be between 1 and N-1, inclusive.

You are given a int[] **scores** with N-1 elements. The score of a node with degree d is **scores**[d-1]. The score of a graph is the sum of the scores of its nodes.

Your method should compute and return the maximum possible score for a graph you can construct.

Definition

Class:	P8XGraphBuilder
Method:	solve
Parameters:	int[]
Returns:	int
Method signature:	int solve(int[] scores)

(be sure your method is public)

Notes

-In your solution, the number of nodes N in your graph can be determined as one plus the length of **scores**.

-In your graph, there must be at most one edge connecting any pair of nodes, and an edge cannot connect a node with itself.

Constraints

-**scores** will contain between 1 and 50 elements, inclusive.

-Each element in **scores** will be between 0 and 10,000, inclusive.

Examples

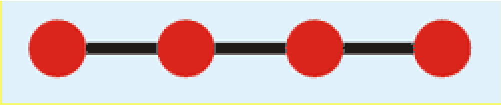
0)

```
{1, 3, 0}

Returns: 8
```

As **scores** contains 3 elements, we are building a graph with N=4 nodes. Nodes of degree 1 have score 1, nodes of degree 2 have score 3, and nodes of degree 3 have score 0.

One possible graph with the highest score looks as follows:



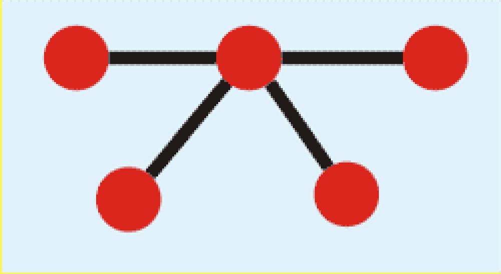
In this graph the degrees of the nodes are 1, 2, 2, 1, respectively. The sum of their scores is 1+3+3+1 = 8.

1)

```
{0, 0, 0, 10}

Returns: 10
```

One possible solution for this test case is:



2)

```
{1, 2, 3, 4, 5, 6}

Returns: 12
```

3)

```
{5, 0, 0}

Returns: 15
```

4)

```
{1, 3, 2, 5, 3, 7, 5}

Returns: 20
```