# Introduction

You can use any of the following languages to perform this task: Java/Kotlin, Go, Scala or C++. Coding best-practices should be present: code comments and unit-tests. Completing the task should be doable in 1.5 hours.

## Introduction: The sawmills in LEGO land

Daily woodcutters of LEGO land cut trees in the mountains. Every tree trunk is precise 1, 2, 3, etc blocks in length. This is LEGO land you know! Trees are mere simple blocks. The weak spot of a tree is the trunk, so woodcutters cut the tree from the ground. Then the leaves are removed. At the end of the day the woodcutters throw the tree trunks in any order they like into the river.

More downstream there is a sawmill to where the tree trunks will float. The river has the same width as the tree trunks and therefore tree trunks cannot overtake each other. So, the first tree trunk the woodcutters throw into the river will arrive first at the sawmill. The space between tree trunks is neglectable. In LEGO land everything fits very precise.

When the woodcutters return from their work, they will turn the sawmill on. Unfortunately, the machine is old, and the speed cannot be regulated anymore. The mechanics are broken, so every 3 blocks it just makes a cut. For example, if the first tree trunk is 4 blocks in length, then it will produce sawn wood of length 3 and a remainder of length 1. If a next trunk is again 4 block in length, then it will produce sawn wood of length of 2 and, a remainder of length 2.

Unfortunately, by the specific building rules in LEGO land, only sawn wood of length 2 is used by the carpenters. To be more precise:

* Sawn wood of length of 1 has a profit of -1. It costs money to get rid of it.
* Sawn wood of length 2 has a profit of +3.
* Luckily, sawn wood of length 3 can still be sold for +1 as it can be shortened by the carpenters themself.

Note: the exercise continues on the next page.

## Exercise

The input describes several test cases:

1. The first line of input for each test case contains a single integer Z, the number of sawmills (each is connected to its own river) in the test case.
2. This is followed by Z lines, each describing the tree trunks.
3. The first number in each line is the number E of tree trunks that are cut.
4. Following it are E strict positive integers, indicating the length (in blocks) of the tree trunks.
5. The input is terminated by a description starting with Z = 0.

This description should not be processed.

For each test case,

1. Print the case number (1, 2, ...).
2. Then print two lines, the first containing the **maximum** profit the woodcutters can achieve.
3. The second line should specify the order the woodcutters should throw the trees into the river between squares. Whereas the first number represents the first tree trunk to be thrown into the river. Idem, the last number is the last tree trunk thrown into the river. If this order is not uniquely determined, print all possible orders. Remove all duplicates.

**Example input**1  
3 2 3 1  
3  
3 1 2 1  
2 1 2  
2 1 4  
0**Example output**Case 1  
Max profit: 4  
Order: [1 3 2] [2 3 1]  
Case 2  
Max profit: 8  
Order: [1 2 1] [2 1 1], [1 2] [2 1], [1 4]

Note! the examples describes the format, not necessarily the correct answer of the exercise.

\* If you are stuck after 30 mins, you can ask for a hint to get you on the right track

\*\* Complete solution should be uploaded to github in a private repository and shared with the following user(s): jmoree-snip