**Energy Pie**

*After battle all of the robots return to base for recharging. But there's a problem with their energy meters and not all all the bots know how much energy to take.*

The robots were returning to the energy charging base in small groups or by themselves and did not see each other upon their return. This meant that each group did not know how many robots had fueled their part of the squad energy container. Some groups knew what the charged capacity of the battery was, but other groups didn't know this information and think the remaining juice is the full capacity of the battery, then take their portion from the remaining electricity. So, each group had divided the remaining or initial charge and took their part. How much power will remain after all of the robots return from their mission?

Let’s take a look at an example of how this should work: There are 6 robots. The first group consists of 2 robots. They had divided the energy into 6 parts and took 2/6 of it. The remainder of the energy is 2/3 of the initial charge. Next returns a single robot, it doesn’t know about the original charge of the battery, so it divides the remaining energy into 6 parts and takes 1 part. This leaves 10/18 = 5/9. The last group is 3 robots, which knew about the original charge. They took half of the original container, so the remaining is 5/9 - 3/6 = 1/18

You are given an ordered array with sizes of the groups in the order they arrived. If a group knows about the initial charge capacity, then the charge is positive. If not, then charge will be negative. The recent example will given as (2, -1, 3).

###### \*\*\*\*\*\* ...... ......

###### \*\*\*\*\*\* ...... ......

###### ###### \*\*\*\*## ....\*\*

###### ###### ###### \*\*\*\*\*\*

###### ###### ###### \*\*\*\*\*\*

###### ###### ###### \*\*\*\*##

 init 2 -1 3

You should calculate the amount as remaining energy in the container as a fraction from the starting size. The result should be represented as two numbers - numerator and denominator. Zero should be represented as (0, 1).

**Input:** Sizes of groups as a tuple of integers.

**Output:** The remaining of the pie as a list or tuple of two integers.

**Example:**

energy\_pie((2, -1, 3)) == (1, 18)

energy\_pie((1, 2, 3)) == (0, 1)

**Precondition:**

∀ x ∈ groups: -1000 < x < 100 AND x ≠ 0

**How it is used:**

This mission will familiarize you with operations involving fractions because we must deal with fractions every day of our lives.

[Repository](https://github.com/Checkio-Game-Missions/checkio-empire-energy-pie.git)

**Digits Multiplication**

*After programming the computers to handle division we should add functionality for multiplication. For this we will use a simple numbers exercise.*

You are given a positive integer. Your function should calculate the product of the digits **excluding any zeroes**.

For example: The number given is 123405. The result will be 1\*2\*3\*4\*5=120.

In this mission the main goal to make your code as short as possible. The shorter your code, the more points you earn. Your score for this mission is dynamic and directly related to the length of your code.

**Input:** A positive integer.

**Output:** The product of the digits as an integer.

**Example:**

golf(123405) == 120

golf(999) == 729

golf(1000) == 1

golf(1111) == 1

**Precondition:**

0 < number < 106

**Scoring:**

Scoring in this mission is based on the number of characters used in your code (comment lines are not counted).

**Rank1:**

Any code length.

**Rank2:**

Your code should be shorter than 100 characters.

**Rank3:**

Your code should be shorter than 70 characters.

**How it is used:**

This mission teaches you to work with simple data type conversions to solve a problem.

[Repository](https://github.com/Checkio-Game-Missions/checkio-empire-digits-multiplication.git)