



Backend Test Task

Congratulations! You have received a contract from NASA for a software application that will help to calculate the fuel required for the flight. This application aims to calculate fuel to launch from one planet of the solar system and land on another planet of the solar system, depending on the flight route.

The formula to calculate fuel is quite simple, but it depends on the planet's gravity. Planets NASA is interested in are:

- Earth - 9.807 m/s^2
- Moon - 1.62 m/s^2
- Mars - 3.711 m/s^2

The formula for fuel calculations for the launch is the following:

$mass * gravity * 0.042 - 33$ rounded down

The formula for fuel calculations for the landing is the following:

$mass * gravity * 0.033 - 42$ rounded down

For example, for the Apollo 11 Command and Service Module, with a weight of 28801 kg, to land it on Earth, the required amount of fuel will be:

$$28801 * 9.807 * 0.033 - 42 = 9278$$

However, fuel adds weight to the ship, so it requires additional fuel until the additional fuel is 0 or negative. Additional fuel is calculated using the same formula from above.

Example:

9278 fuel requires 2960 more fuel
2960 fuel requires 915 more fuel
915 fuel requires 254 more fuel
254 fuel requires 40 more fuel
40 fuel requires no more fuel

So, to land Apollo 11 CSM on the Earth, 13447 fuel is required ($9278 + 2960 + 915 + 254 + 40$).

The application should receive a flight route as 2 arguments. The first one is the flight ship mass, and the second is an array/list of 2 element tuples, with the first element being `land` or `launch` directive, and the second element is the target planet gravity.

But consider that to land a module on the Moon, you need additional fuel, which should be launched from the Earth - we don't have a refuel station in space - and we need to carry fuel for all steps from the beginning.

For example, for the program to launch the ship from the Earth, land it on the Moon, and return to the Earth, input arguments will look like this:

Ruby - `28801, [[:launch, "earth"], [:land, "moon"], [:launch, "moon"], [:land, "earth"]]`
Elixir - `28801, [[:launch, "earth"], {[:land, "moon"], [:launch, "moon"], {[:land, "earth"]}]`
Python - `28801, [(:launch, "earth"), (:land, "moon"), (:launch, "moon"), (:land, "earth")]`

Additionally, the flight path should be validated, and the program should skip calculations and emit an error stating that gravity is unknown.

And remember, you are hired by NASA, and reliability is crucial. We have no right to make a mistake.

Here are examples of programs and required fuel for the whole mission:

1. Apollo 11:
 - path: launch Earth, land Moon, launch Moon, land Earth
 - weight of equipment: 28801 kg
 - weight of fuel: 51898 kg
2. Mission on Mars:
 - path: launch Earth, land Mars, launch Mars, land Earth
 - weight of equipment: 14606 kg
 - weight of fuel: 33388 kg
3. Passenger ship:
 - path: launch Earth, land Moon, launch Moon, land Mars, launch Mars, land Earth
 - weight of equipment: 75432 kg
 - weight of fuel: 212161 kg

P.S. Creating working, maintainable, and readable codes is essential, so take your time and do your **best**.

P.P.S. Please create a simple project without the usage of web frameworks.

! Please upload your solution to GitHub and provide a link.



Additional Guidelines for Elixir

While it's possible to create a working solution that doesn't take the following points into account, successful implementations that conform to the following will be given special consideration:

- 1) Prefer a functional style. Specifically with Elixir idioms such as piping and avoiding intermediate variables unless required for clarity.
- 2) Where possible and in good style, make use of pattern matching.
- 3) Treat the fuel calculation logic as an external service implemented as a GenServer.
- 4) For extra consideration and if time permits, use a Supervisor to launch the GenServer mentioned in guideline three and explain the circumstances under which a Supervisor would restart a crashed GenServer and how many restarts would be attempted.