Go Day08

Here in a jungle you can find some weird creatures that you need to treat in an unusual way. For this task you need to write a function getElement(arr []int, idx int) (int, error) that accepts and an index and gives you back the element with this index. Seems easy enough, eh? But here’s one condition - you can’t use lookup by this index (like arr[idx]), the only lookup allowed is a first element (arr[0]). You may need to remember some C to complete this exercise.  
In case of any non-valid input (empty slice, negative index, index is out of bounds) the function should return an error with a text explanation of a problem.

[14:59](https://21born2code.slack.com/archives/D04CPPLS9BR/p1670673572845639)

You’re in luck! You’ve found some pretty rare plants:

type UnknownPlant struct {

FlowerType string

LeafType string

Color int `color\_scheme:"rgb"`

}

type AnotherUnknownPlant struct {

FlowerColor int

LeafType string

Height int `unit:"inches"`

}

Well, yeah, current representation is a bit of a mess. Your goal would be to write a single function describePlant that will accept any kind of plant (yes, it should work with structures of different types) and then print all fields as key-value pairs, separated by comma (mind the tags), like this:

FlowerColor:10

LeafType:lanceolate

Height(unit=inches):15

**Новое**

[14:59](https://21born2code.slack.com/archives/D04CPPLS9BR/p1670673583720579)

Okay, now it’s time to relax and have some cocoa. Cocoa usually comes in packages (see provided zip archive). You don’t need to modify the code in packaged files in any way, the only thing you need to do is write a code (including cocoa files as part of your project) that will create default empty Mac OS GUI window (size 300x200) with title “School 21”. It’s easier than you think!