Project	3
Due	Noon, Dec. 3

This project uses methods and interfaces. Note that interface{} is the empty interface. Interfaces are the way that Go provides abstract types.

You are to write a single program, container.go.

1.1 Container and Iterator interface types

You are to define Container and Iterator interface type. Note that together with the interface{}, these interfaces contain the abstract type.

Containers The Container has the following methods:

- begin() Iterator
- append(v interface{})

Iterator The Iterator has the following methods

- Next() Iterator
- IsEnd() bool
- Deref() interface{}

1.2 Vector and List

You should implement two Container types and their associated Iterators. The below types are the concrete types which implement the abstract (interface) types.

- Vector which works like a C++ vector
- List which is a linked list

1.3 SumInt and SumFloat

To use the interfaces, you should write two functions:

- SumInt(c Container) int which sums all the elements which are ints in the container.
- SumFloat64(c Container) float64 which sums all the elements which are float64s in the container.

2 Hints

Here are some preliminaries you should master before doing the project

- Assigning an int to interface{}.
- Find out whether the concrete type underlying the empty interface is an int or an float64.
- extract an int from interface{}
- Writing an interface with one method, and a concrete type and method which conforms to that interface

Remember that

- to search on the web for Go use the term "golang".
- the go playground is very useful for testing out small snippets of code.