Rust SEP Winter Term 25/26
Exercise 2: Ownership and Borrowing

Task 1: Ownership Quiz

a) Solve the Moodle quiz about ownership and borrowing!

Task 2: Fixing Ownership Errors

a) Fix the functions in ex02!

   Use the compiler/rust-analyzer messages to help you!

Task 3: Linked Lists

Consider the following struct definition for a singly linked list:

```
struct LinkedList {
  // Rust's `Option` == Haskell's `Maybe`
  head: Option<Box<Node>>,
}

struct Node {
  value: i32,
  next: Option<Box<Node>>,
}
```

a) Draw the ownership graph for a list with a length of 3. The source of an edge is the owner and the target is the owned value.
b) Can you guess why the `next` field uses `Box<Node>` instead of just `Node`? If not, just try it out!

Now consider a doubly linked list with the following definition:

```
struct DNode {
  value: i32,
  next: Option<Box<DNode>>,
  prev: Option<Box<DNode>>,
}
```

c) Draw the ownership for a double-linked list with a length of 3
d) Why isn't it possible to write a double-linked list implementation for the given struct `DNode`?
e) Can you think of another data structure with the same problem?
f) Can you generalize the problem?

Task 4: Continue Coding

Now that you have learned about ownership and borrowing, you are ready to write some non-trivial programs in Rust.