```
/* pointers and structs */

// here MyStruct has a field that is a pointer to a u32 value
struct MyStruct {
  ptr_field u32*
}

instance MyStruct

// a pointer to a MyStruct value
ptr_mystruct MyStruct* = &instance

// create a MyStruct value on the heap
heap_mystruct MyStruct* = alloc(MyStruct) {
  .ptr_field = ptr_mystruct
}


/*
 * if we defer heap_mystruct the pointer
 * inside the instance will be deleted.
 */
defer heap_mystruct


/*
 * if we don't want to delete the pointer inside heap_mystruct
 * then we need to modify struct MyStruct to tell the compiler
 * it doesn't own the data pointed by ptr_field
 */

struct MyStruct {
  !ptr_field u32*
}

heap_mystruct MyStruct* = alloc(MyStruct) {
  .ptr_field = ptr_mystruct
}

/*
 * now if we defer heap_mystruct
 * the pointer inside the instance
 * will not be deleted.
 */
defer heap_mystruct
```