Student's Question: I don't understand these two lines of code:
`c = c[:1+copy(c[1:], c[2:])]`

`d = d[:len(d)-1]`

The explanation was too quick for me. Could you please give us more examples or do it step-by-step?
I read the link described in the comments section but didn't find an elaborated explanation either.

Answer - First part of the question:

```
x := []int{1, 5, 6, 7}

x1 := x[1:]           //[5 6 7]
x2 := x[2:]           //[6 7]
x3 := x[:1]           //[1]

fmt.Printf("\nx=%v, x1=%v x2=%v x3=%v", x, x1, x2, x3)

// Don't forget that slices are not actual copies.
// They point to the base array, therefore the following
// copy() is changing the base array by copying 6,7 starting
// at index 2 (as x2 suggests) to the starting index that x1
// suggests (that is index 1) and that's why 5,6
// has changed to 6,7 (index 1 and 2).
//
x4 := copy(x1, x2)
fmt.Printf("\nx=%v, x1=%v x2=%v x3=%v x4=%v", x, x1, x2, x3, x4)

// From the previous line x4 returned 2, therefore the
// following is like x5 := x[:3] ,
// meaning from index 0 to 2 (inclusive)
//
x5 := x[:1+x4]
fmt.Printf("\n\nx5=%v", x5)
```

output:

```
x=[1 5 6 7], x1=[5 6 7] x2=[6 7] x3=[1]
x=[1 6 7 7], x1=[6 7 7] x2=[7 7] x3=[1] x4=2

x5=[1 6 7]
```

Answer - Second part of the question:

```
d := []int{1, 2, 3, 4, 5, 6, 7}
i := len(d) - 1 //number of of elements od d, minus 1
// fmt.Printf("i=%v i=%v\n", i, d[i]) //6

fmt.Printf("i=%v, d=%v d[2]=%v d[i]=%v\n", i, d, d[2], d[i])

// d[i] is like d[len(d)-1]
d[2] = d[i]
fmt.Printf("i=%v, d=%v d[2]=%v d[i]=%v\n", i, d, d[2], d[i])
```

output:

```
i=6, d=[1 2 3 4 5 6 7] d[2]=3 d[i]=7
i=6, d=[1 2 7 4 5 6 7] d[2]=7 d[i]=7
```