

Aufgabe 1: Syntax von Programmen**(10 Punkte)**

Welche der folgenden Variablendeklarationen sind syntaktisch korrekt? Bei den nicht korrekten Zeilen erläutern Sie jeweils kurz den Fehler.

```
1    x := 42
2    var int y 55
3    int z = 42
4    s := string([]byte{'a', 'b', 'c'})
5    b := []byte{'a', 'b', 'c'}
6    var l1 []int := make([]int, 0)
7    string := hallo
```

Aufgabe 2: Fehlersuche: Syntaxfehler**(10 Punkte)**

Das folgende Programm enthält eine Reihe an Syntaxfehlern, durch die es nicht compiliert. Markieren Sie alle Zeilen, die einen Fehler enthalten und erläutern Sie kurz, was jeweils falsch ist.

```
1 package foo
2
3 import "fmt"
4
5 func PrintSomething(what string) string {
6     fmt.Print(what)
7     fmt.Print("\n")
8 }
9
10 func ComputeProduct(numbers int) int {
11     result := 1
12     for _, num := range numbers {
13         result *= num
14     }
15     return result
16 }
17
18 func main() {
19     p = ComputeProduct(1, 3, 5, 2, 0, 2)
20     PrintSomething(string fmt.Sprintf(p))
21 }
```

Hinweis: Es geht hier nicht um inhaltliche Fehler, nur um Syntaxfehler.

Anmerkung: Für jede falsch markierte Zeile gibt es Punktabzug!

Aufgabe 3: Fehlersuche: Inhaltliche Fehler**(10 Punkte)**

Die folgende Funktion ist zwar syntaktisch korrekt, sie erfüllt aber nicht ihre Aufgabe. Erläutern Sie den/die Fehler und machen Sie einen Vorschlag zur Korrektur.

```
1 // IsPrime liefert true, falls n eine Primzahl ist.
2 func IsPrime(n int) bool {
3     for i := 2; i < n-1; i++ {
4         if n%i == 0 {
5             return false
6         } else {
7             return true
8         }
9     }
10    return true
11 }
```

Anmerkung: Ihr Korrekturvorschlag muss kein syntaktisch korrekter Code sein. Eine Erklärung in Worten genügt.

Aufgabe 4: Programmverständnis**(10 Punkte)**

Erläutern Sie, was die folgende Funktion berechnet. Geben Sie eine möglichst allgemeine bzw. abstrakte Erklärung an. Erklären Sie auch, mit welcher Art von Argumenten diese Funktion sinnvoll arbeitet.

```
1 func Foo(m, n int) bool {
2     if m == 0 || m > n {
3         return false
4     }
5     if n == m {
6         return true
7     }
8     return Foo(m, n-m)
9 }
```

Aufgabe 5: Datenstrukturen**(10 Punkte)**

Entwerfen Sie eine Datenstruktur, die ...

Hier Aufgabe einsetzen!