

GRA-Übungsblatt 02

1. a, b) Siehe ML

- c)
- v0: globale Variable, initialisiert und schreibbar → .data
 - v1: globale Variable, nicht initialisiert und schreibbar → .bss
 - v2: globale Variable, initialisiert, nicht schreibbar (const) → .rodata
 - v3: lokale Variable → Stack
 - v4: lokale Variable → Stack
 - v5: Allokation mittels malloc → Heap
 - v6: Allokation mittels alloca → Stack

2. a)
- `array[i]` ist ein normaler Zugriff auf das Element `i` im Array (wie bspw. in Java)
 - `*array` ist die Deklaration von 'array'. Der Arraybezeichner selbst verhält sich wie ein Pointer auf das erste Element des Arrays
 - `*(array + i)` entspricht `array[i]`. Zum "Pointer" 'array' wird `i` dazugaddiert; nach Regeln der Pointerarithmetik entspricht das dem Inkrement der Adresse um `i · Größe des Datentyps`, auf den der Pointer zeigt (bspw. 4 Byte für int, abh. von der Zielarchitektur).
 - `&array[2]` entspricht `&*(array + 2)`. Wir wählen das Element 2 des Arrays und holen uns davon mit `&` die Adresse.

3. a) Siehe ML