

## DS – Übung 09

### U 9.1)

Gegeben ist folgender Code:

```
-----  
.section .data  
aSh:  
    # Array 3 x 4  
    .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12  
  
.section .text  
.globl main  
.type main, @function  
main:  
    pushq %rbp  
    movq %rsp, %rbp  
  
    # aSh in rdi, i in rsi, j in rdx  
    # interim results in rax and rcx  
    # Add 5 and 12 from the array  
  
    # exit main  
    movq $0, %rax  
    popq %rbp  
    ret  
-----
```

Addieren Sie die Zahlen 5 und 12 aus dem Array auf die gleiche Weise und gehen Sie dabei genauso vor wie im Skript **Kap.9, Arrays, S.13 - 15**. Achten Sie auf den **Datentyp** und die **Größe** des Arrays 3 x 4!

### U 9.2)

Gegeben ist folgende Struktur:

```
struct St{  
    char c;  
    int n;  
    int *pN;  
};
```

Schreiben Sie ein Assembler-Programm, das drei Objekte einer solchen Struktur erzeugt. **pN** soll dabei die Adresse von **n** beinhalten.

- Objekt 1:  
Verwenden Sie "minimal gepackte Datenstruktur".
- Objekt 2:  
Verwenden Sie "natural alignment".
- Objekt 3:  
Verwenden Sie "Optimieren des Alignments durch Umordnung".

Geben Sie anschließend die Summe der drei int-Werte aus den verschiedenen Strukturen aus, indem Sie über die jeweiligen Zeiger **pN** darauf zugreifen.