

## U 11.1)

Gegeben sind folgende vier Ausdrücke:

1.  $13.99 - 17.4$
2.  $\sqrt{2.0} + \sqrt{3.0}$
3.  $-214.0 * 0.75$
4.  $\max(5.5, 30.52)$

Übersetzen Sie folgende C-Funktion in GNU-Assembler:

```
// truncates decimals
long truncDec(float * f){
    float sum = 0.0;
    for(int i = 0; i < 4; i++)
        sum += f[i];
    return sum;
}
```

Verwenden Sie **keine ymm-**, sondern nur **xmm- und General Purpose-Register**.

Geben Sie das Ergebnis mittels `printf("Sum = %ld\n", ...)` aus.

Das Ergebnis lautet: **-130**

-----  
Folgende Struktur ist Ihnen für die Lösung bereits vorgegeben:

```
.section .rodata
format:
.asciz "Sum = %ld\n"
.section .bss
.lcomm results 16

.section .data
.align 16
farr1:
.float 13.99, 17.4, 2.0, 3.0
.align 16
farr2:
.float -214.0, 0.75, 5.5, 30.52

.section .text
.globl main
.type main, @function
main:
    # ...
    # calc 1. expression
    # ...
    # calc 2. expression
    # ...
    # calc 3. expression
    # ...
    # calc 4. expression
    # ...
```

```
# call truncDec
leaq results, %rdi
call truncDec

# call printf
leaq format, %rdi
movq %rax, %rsi
movq $0, %rax
call printf

# exit main
movq $0, %rax
popq %rbp
ret

.globl truncDec
.type truncDec, @function
truncDec:
# ...
ret
```