

## Aufgabe 1

```
// our working number
volatile uint8_t number = 1;
// baudrate for serial communication
const int baudRate = 9600;
// initialize Serial
void setup() {
    Serial.begin(baudRate);
}
// main loop
void loop() {
    // print number
    Serial.println(number);
    asm volatile(
        //TODO: insert program here
        "mov r4, %[value]\n\t"
        "cmp r4, #128\n\t"
        "ite eq\n\t"
        "moveq r4, #1\n\t"
        "addne r4, r4\n\t"
        "mov %[value], r4\n\t"
        : [value] "r+" (number)
        :
        : "r4", "memory", "cc"
    );
}
```

Operation	Taktzyklen
mov	1
cmp	1
ite	1
moveq	1
addne	1
mov	1

Insgesamt sind es bei Aufgabe 1 also 6 Taktzyklen.

## Aufgabe 2

```
// our working data
```

```

const uint8_t lastFiboIndex = 13;
uint8_t fibData[lastFiboIndex];
const int waitTime = 1000;
// baudrate for serial communication
const int baudRate = 9600;
void setup() {
    // initialize Serial
    Serial.begin(baudRate);
    // init first two Fibonacci numbers
    fibData[0] = 1;
    fibData[1] = 1;
    asm volatile(
        // TODO: insert program here

        "mov r4, %[fibData]\n\t"
        "loopFibonacci:\n\t"
        "ldr r5, [r4, #0]\n\t"
        "ldr r6, [r4, #1]\n\t"
        "add r6, r5\n\t"
        "str r6, [r4, #2]\n\t"
        "add r4, #1\n\t"
        "ldr r6, [r4, #0]\n\t"
        "add r7, #1\n\t"
        "cmp r7, %[index]\n\t"
        "it ne\n\t"
        "bne loopFibonacci"
        :
        : [fibData] "r" (fibData), [index] "r" (lastFiboIndex)
        : "r4", "r5", "r6", "r7", "cc", "memory"
    );
}
// main loop
void loop() {
    // print data
    for (int i = 0; i < lastFiboIndex; i++)
    {
        Serial.println(fibData[i]);
    }
    // delay 1s
    delay(waitTime);
}

```

Operation	Taktzyklen
mov	1
ldr	2
ldr	2
add	1
str	2
add	1
ldr	2
add	1
cmp	1
it	1
bne	2

Bei Aufgabe 2 sind es insgesamt 16 Taktzyklen.