

Test 1 nachtragen!!!

1. Die vorzeichenlose Zahl in r0 soll durch 16 geteilt werden. Das Ergebnis soll in r1 stehen. Geben Sie den Befehl an.

Lösung: `mov r1, r0, LSR#4`

2. Das .data-Feld beginne bei Adresse 0x2000. Welcher Wert (Hex.) steht für Var1 in der Symboltabelle? (Erst nachdenken!)

`.section .data`

`Tab: .word 0x10, 'A', 10, '1'`

Lösung:

0x2000	0x2001	0x2002	0x2003
10	00	00	00

3. Folgendes Datenfeld sei gegeben:

`.section .data`

`Var1: .word 0x10, 0xAA12`

Geben Sie die Assemblerbefehle an, um das **zweite Datenwort** des Feldes nach r1 zu kopieren.

Lösung: `ldr r0, = Var1`
`ldr r1, [r0, #4]`

4. Was steht in r0 nach folgendem Befehl (Hex.)?

`ldr r0, = 0x1234ABCD`

Lösung: `r0 = 1234ABCD`

5. In welchem **Wertebereich** muss r0 liegen, damit ein Sprung nach LOOP erfolgt?

`cmp r0, #15`

`bhi LOOP`

Lösung: Größer oder gleich: `16d`
 Kleiner oder gleich: `0x F...F`

6. In welchem **Wertebereich** muss r0 liegen, damit ein Sprung nach LOOP erfolgt?

`mov r1, #-15`

`cmp r0, r1`

`bge LOOP`

Lösung: Größer oder gleich: `-15`
 Kleiner oder gleich: `0x7F...F`

7. In welchem **Wertebereich** muss r0 liegen, damit ein Sprung nach LOOP erfolgt?

`ldr r1; =0xFFFFF80`

`ands r0, r1`

`beq LOOP`

Lösung: Größer oder gleich: `0`
 Kleiner oder gleich: `0x7F`

Gegeben ist folgendes Programmfragment (für Aufgabe 1-3)

```
.section      .data

VarA:  .word      30, 65, 0xA, 'A'

VarB:  .byte      33, -1, 0x10, 0x20, 0x30, 'a', '1'
```

Aufgaben

1. Mit welcher Befehlsfolge (4 Befehle) addiert man die ersten beiden Werte von VarA?

Lösung:

```
ldrb r1, = VarA
ldrb r0 [r1]
ldrb r2 = VarA
ldrb r1 [r2]
add r2, r1, r0
```

2. Das .data-Feld VarA beginnt bei Adresse 0x2000. Geben Sie die Speicherinhalte (Hex.) an den folgenden Adressen aus.

Lösung:

0x200c	0x200D	0x200E	0x200F	0x2010	0x2011
41	00	00	00	11	FF

3. Mit welchem Befehl legt man den Inhalt von Register r0 auf den Stack ab (push)?

Lösung: `Stmfd sp!, {r0}`

4. Mit welchem Befehl springt man zu einem Unterprogramm „MyProg“?

Lösung: `bl MyProg`

5. Wozu dient das Linkregister?

Lösung: Als Rücksprungadresse zwischen Haupt- und Unterprogramm. Also merkt sich die Stelle, wo das mainprog verlassen wird und später von nach ausführung von MySubroutine wieder zurückspringt

6. Wozu dient der Framepointer?

Lösung: Über den Framepointer fp kann man bequem auf den lokalen Speicher zugreifen! Sprich Lokale Daten auf Stack.

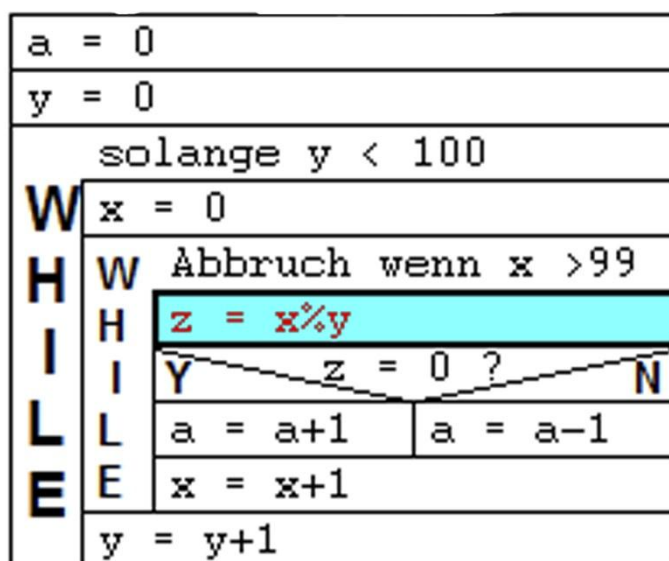
7. Geben Sie einen Befehl an, mit dem folgendes berechnet wird: $[r0] \leftarrow [r0] + 4 * [r1]$.

Lösung: `Add r0, r1, LSL#2`

8. Geben Sie eine Befehlssequenz an, die zum Label „TestOK“ springt, wenn die Bits 0 und 3 in r0 gesetzt sind.

Lösung:

1. Vereinbaren und initialisieren Sie mit Hilfe eines Feldes einen String mit dem Inhalt „Hallo“;
Lösung: `char txt[]="Hallo";`
2. Vereinbaren und initialisieren Sie mit Hilfe eines Pointers einen String mit dem Inhalt „Hallo“;
Lösung: `Char *txt="Hallo";`
3. Initialisieren Sie den Zeiger px so, dass er auf den Vektor x zeigt.
`int x[] = (1, 2, 3);`
`int *px;`
Lösung: `int *px = x;`
4. `int Val = 25, ValCopy;`
`int *pVal;`
 - a) Initialisieren Sie den Zeiger pVal so, dass er auf die Variable Val zeigt.
Lösung: `int*pVal = &Val`
 - b) Kopieren Sie mit Hilfe des Zeigers pVal den Wert auf den dieser Zeiger zeigt nach ValCopie.
Lösung: `ValCopy = *pVal`
5. Schreiben Sie ein C-Programm(-fragment) zum abgebildeten Struktogramm.



```

int main() {
    int a = 0;
    int y = 0;
    while (y<100) {
        int x = 0;
        while (x <= 99) {
            z= x%y;
            if (z==0){
                x++;
            }
            y++;
        }
        return 1;
    }
}

```