

Übung 4 - Lösung

1. Multiplikation mittels fortgesetzter Addition

Erstellen Sie ein Programm, das zwei Zahlen mittels fortgesetzter Addition miteinander multipliziert, d.h. z.B. 5 mal 3 ist 5 plus 5 plus 5.

Die Multiplikatoren sind 32-Bit Werte. Das Ergebnis wird in einer 8 Byte Variablen PROD abgespeichert ($PROD = MULTA * MULTB$)

- a) Erstellen Sie ein Flussdiagramm für den Algorithmus
- b) Erstellen Sie das Assembler Programm
- c) Führen Sie die Multiplikation mit folgenden Werten durch:

MULTA	MULTB	PROD
255 h	255 h	5 7039 h
FFFF h	FFFF h	FFFE0001h
1 0002 h	FFFF h	1 0000 FFFE h
AB CDEF h	AB CDEF h	734C C2F2 A521 h
AB CDEF h	FFFF FFFF h	AB CDEE FF54 3211 h

ACHTUNG:

Wenn Sie das Ergebnis mit PRINT_HEX ausgeben, so wird bei 00ABCDEFh nur ABCDEF ausgegeben!

Besser ist es, die Variablen direkt zu Überwachen. Hier aber bei PROD den Typ Hex und q verwenden!

```
%include "io.inc"
```

```
section .data
```

```
MULTA DD 00000000h  
MULTB DD 00000000h  
PROD DQ 0000000000000000h
```

```
section .text
```

```
global CMAIN
```

```
CMAIN:
```

```
    mov ebp, esp; for correct debugging  
    ;write your code here
```

```
    GET_HEX 4, [MULTB]  
    GET_HEX 4, [MULTA]
```

```
    ; RECHNEN
```

```
    mov EAX,0
```

```
    mov EDX,0
```

```
    mov ECX,0
```

```
    mov ECX,[MULTB] ;schreibe Multiplikator in CX
```

```
    Schleife:
```

```
    add EAX,[MULTA] ;Addiere Multiplikand zu AX dazu
```

```
    jnc KeinUebertrag ;kein Übertrag -> überspringe Übertragsaddition
```

```
    add EDX,1 ;Übertragsaddition: EDX+=1
```

```
    KeinUebertrag:
```

```
    Loop Schleife ;bei CX!=0 springe zu Schleife
```

```
    mov DWORD [PROD],EAX ;verschiebe ersten Teil des Ergebnisses in PROD
```

```
    mov DWORD [PROD+4],EDX ;verschiebe zweiten Teil des Ergebnisses in PROD+4
```

```
    PRINT_HEX 4, [PROD+4]
```

```
    PRINT_HEX 4, [PROD]
```

```
    xor eax, eax
```

```
    ret
```