

# Übung 3

## 1. (A - B) + C

Entwickeln Sie ein Assemblerprogramm, das für verschiedene (32-Bit) Werte von A, B und C den folgenden Ausdruck berechnet:

$$\text{ERG} = (A - B) + C$$

Die Größen A, B und C sollen anschließend jeweils verdoppelt werden und die Berechnung soll erneut durchgeführt werden.

Falls bei der Berechnung ein „Overflow“ auftritt, soll das Programm beenden und der Betrag einer Zählvariablen, die die „Durchläufe“ mitzählt, soll in der Speicherzelle „Zahl“ sowie das Ergebnis ERG in einer Variablen „ERG“ abgespeichert werden. Außerdem sollen die Werte für A, B und C beim Abbruch gespeichert werden.

- Erstellen Sie ein Flussdiagramm für den Algorithmus
- Erstellen Sie das Assembler Programm
- Füllen Sie die folgende Tabelle mit Hilfe des Programms aus:

A Start	B Start	C Start	A Ende	B Ende	C Ende	ERG	Zahl
1	2	3					
2	3	1					
23h	87h	5h					
5	0	3					

Folgende Codevorlage kann verwendet werden:

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

```
section .data
```

```
    A DD 1h  
    B DD 2h  
    C DD 3h  
    ERG DD 0  
    Zahl DD 0
```

```
section .text
```

```
global CMAIN
```

```
CMAIN:
```

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

```
    PRINT_STRING "A: "
```

```
    PRINT_HEX 4,A
```

```
    NEWLINE
```

```
    PRINT_STRING "B: "
```

```
    PRINT_HEX 4,B
```

```
    NEWLINE
```

```
    PRINT_STRING "C: "
```

```
    PRINT_HEX 4,C
```

```
    NEWLINE
```

```
    PRINT_STRING "Ergebnis: "
```

```
    PRINT_HEX 4,ERG
```

```
    NEWLINE
```

```
    PRINT_STRING "Zahl: "
```

```
    PRINT_HEX 4,Zahl
```

```
    xor eax, eax
```

```
    ret
```