Tutorium 5

1a) Sprung an "marke1" gdw. AL ≥ 5

```
CMP AL,5
JGE marke1
```

```
CMP AL, 4
JG marke1
```

```
SUB AL, 5
JGE marke1
```

```
MOV BL,5
CMP AL,BL
JGE marke1
```

```
CMP AL, 5
JG marke1
JE marke1
```

```
CMP AL, 5
JL marke2
JMP marke1
marke2
```

b)

```
int ebx;
ebx = 50

while((ebx <= 60)) {
    fkt(ebx);
    ebx = ebx + 1;
}
```

Initialisierung des Zählers

Schleifenbedingung

Schleifenrumpf

Zählerfortschaltung

In ASM:

```
        MOV EBX, 50
        JMP cond
body:
        MOV EAX, EBX
        CALL fkt
        INC EBX
cond:
        CMP EBX, 60
        JLE body
```

c)

```
MOV EAX, 100
loop:
    CALL fkt
    DEC EAX
    CMP EAX, -1
    JNE loop
```

ohne CMP mit JL

d)
```
MOV EAX, 0
outer_loop:
   MOV EBX, 0
inner_loop:
    ....
    INC EBX
    CMP EBX, EAX
    JLE inner_loop
    INC EAX
    CMP EAX, 100
    JLE outer_loop
```

2)
```
strcpy:
MOV AL, [ESI]
MOV [EDI], AL
INC ESI
INC EDI
CMP AL, 0
JNE strcpy
RET
```

3a)
```
MOV EBX, EAX
SHR EBX, 3
AND EBX, 0xf
```

b)
```
PUSH EAX
AND EAX, 0x7 (0111)
SHL EAX, 5
AND EBX, 0xffffff1f
OR EBX, EAX
POP EAX
```

c)
```
AND EAX, 0x7800
CMP EAX, 0x4800
JE irgendwo
```

0x7800 = 0111.1000.0000.0000
0x4800 = 0100.1000.0000.0000

d)
```
SAR EAX, 4
```

e)
```
AND EAX, 63
```

siehe ÜB zu Modulo

f)
```
MOV CL, BL
MOV EBX, 1
SHL EBX, CL
NOT EBX
AND EAX, EBX
```

← Nur CL als Register bei SHL