Friedrich-Ebert-Schule Esslingen	**MIKROCONTROLLER**	Name:
5.1	Einlesen / Ausgeben von Bitkombinationen	Datum:

Erstelle nacheinander die Programme 1.1 – 1.5. und übertrage sie auf den Controller.

1.1 Schalterstellungen (Port1) an LEDs (Port2) übertragen

```
code at 0                          ; Programm beginnt bei Adresse 0000h
loop:     mov 0A0h,90h             ; 0A0h Adresse Port 2, 90h Adresse Port 1
          sjmp loop
```

1.2 Schalterstellungen (Port1) an LEDs (Port2) übertragen

```
code at 0
include AT89C5131.INC              ; Datei mit Adressen einbinden
loop:     mov p2,p1                ; Schalterstellungen an LEDs übertragen
          sjmp loop
```

1.3 Schalterstellungen (Port1) an LEDs (Port2) übertragen

```
code at 0
include AT89C5131.INC              ; Datei mit Adressen einbinden
          LEDs      data p2        ; Symbolzuweisung
          Schalter  data p1        ; Hinweis: Statt data ist auch equ möglich,
                                   ; data = Definition als spezielles Funktionsregister
                                   ; equ = Platzhalter, normalerweise Konstante.
loop:     mov       LEDs,Schalter  ; Schalterstellungen an LEDs übertragen
          sjmp      loop
```

1.4 Konstanten an die LEDs ausgeben

```
code at 0
include AT89C5131.INC              ; Datei mit Adressen einbinden
          LEDs      data p2        ; Symbolzuweisung
```

```
          mov       LEDs,#34h      ; Hexzahl ausgeben
loop:     sjmp      loop           ; Endlosschleife
```

1.

```
          mov       LEDs,#34       ; Dezimalzahl ausgeben
loop:     sjmp      loop
```

2.

```
          mov       LEDs,#001101000b   ; Binärzahl ausgeben
loop:     sjmp      loop
```

3.

Teste die 3 Programmteile nacheinander! Wie ändert sich die Ausgabe?

1.5 Einzelne Bit ansprechen (Bitadressen - Byteadressen)

setb bit	= set bit	= Bit setzen (zu 1 machen)
clr bit	= clear bit	= Bit löschen (zu 0 machen)

1. Möglichkeit			2. Möglichkeit			
loop:	**mov**	p2,#0	loop:	**mov**	0A0h,#0	;A0h ist die Byte-Adresse aller 8 Bit von Port2
	setb	p2.3		**setb**	0A3h	;A3h ist die Bit-Adresse von Bit 3 des Port2
	clr	p2.3		**clr**	0A3h	;
	setb	p2.0		**setb**	0A0h	;A0h ist die Bit-Adresse von Bit 0 des Port2
	sjmp	loop		**sjmp**	loop	

(Programm auch mit Einzelschrittausführung im Debugger testen)

=> *Die Unterscheidung, ob es sich um eine Byte-Adresse oder Bit-Adresse handelt, wird durch die Art des Befehles getroffen:* ***MOV*** *oder* ***SETB***