

Øving 6 løsningsforslag

1. Løsningsalternativ A: Tabell ligger i SRAM

seg7:

```
ldi xl,low(tabell)
ldi xh,high(tabell)
clr r22
add xl,r16
adc xh,r22
ld r17,x
ret
```

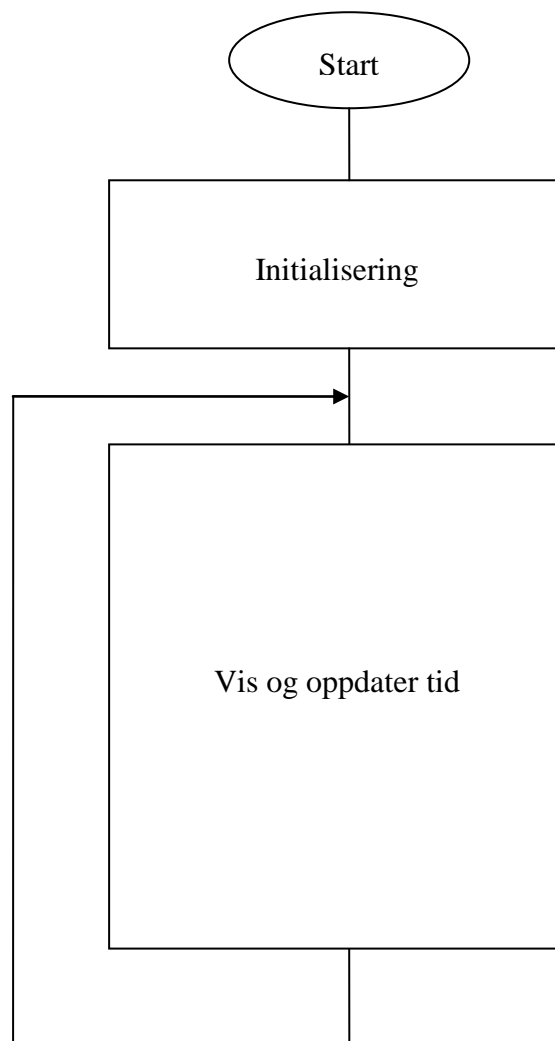
Løsningsalternativ B: Tabellen ligger i flash-minnet

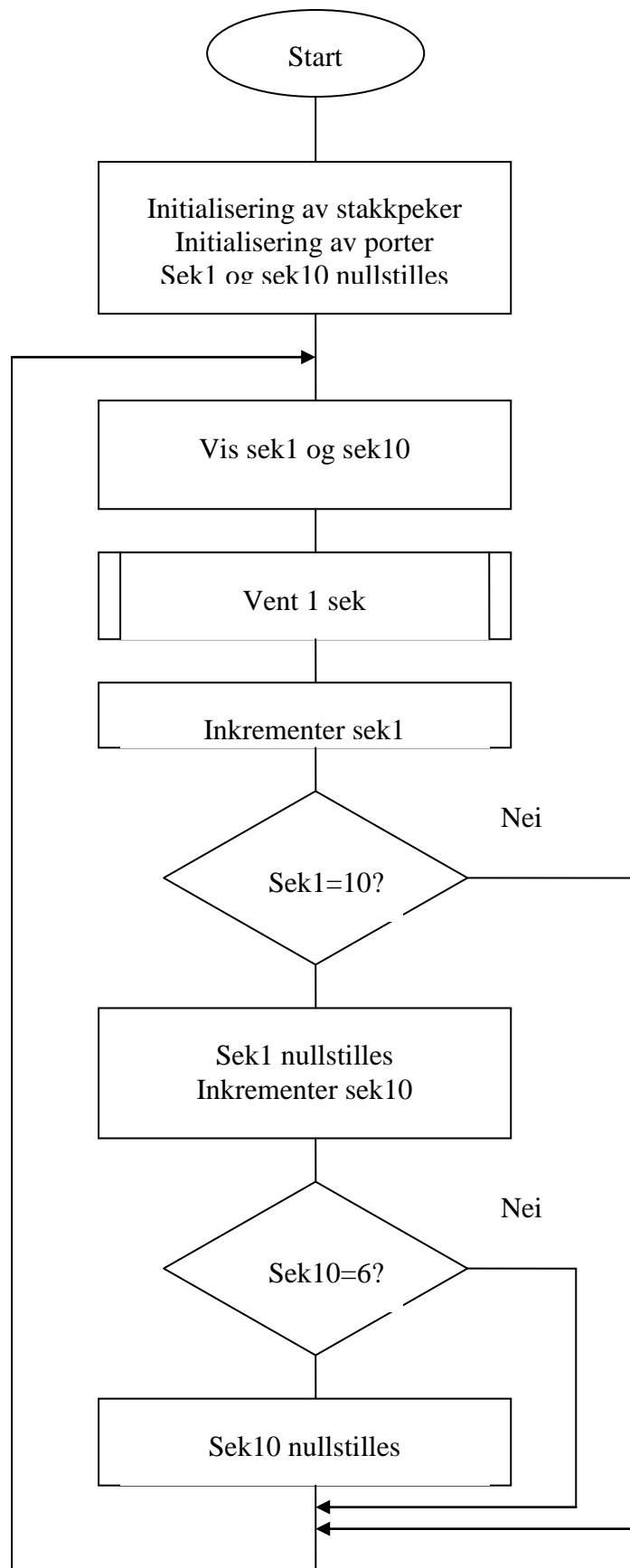
seg7:

```
ldi zl,low(tabell*2)
ldi zh,high(tabell*2)
clr r22
add zl,r16
adc zh,r22
lpm r17,z
ret
```

2. Flytskjema, se neste side.
3. Det vil være naturlig å legge tabellen i Flash-minnet, men det er fullt mulig å bygge opp en tabell i SRAM som vist i nedenfor. Vi må da skrive instruksjoner som legger riktig innhold i de ulike SRAM adressene. Når må vi så bruke SRAM? Tenk på en situasjon hvor vi mottar data fra omverdenen og disse lagres i SRAM. Vi skal lese ut femte data som vi mottok. Da må vi kjenne til hvordan vi leser i SRAM. Legg også merke til hvordan en sikrer riktig innhold i x- og z-register i prosedyrene ovenfor (se på bruken av r22).

Grunnleggende styreprogram-struktur





;Alternativ A: SRAM inneholder tabell

```

.nolist
.include "m128def.inc"
.list
.def sek1=r20
.def sek10=r21
.def kode=r17
.def tall=r16
.def ventetid=r18
.def temp=r19
.org 0
                rjmp start
.org 0x46
start:  ldi temp,low(ramend)
                out spl,temp
                ldi temp,high(ramend)
                out sph,temp
                ldi temp,0x7e
                sts tabell,temp
                ldi temp,0x30
                sts tabell+1,temp
                ldi temp,0x6d
                sts tabell+2,temp
                ldi temp,0x79
                sts tabell+3,temp
                ldi temp,0x33
                sts tabell+4,temp
                ldi temp,0x5b
                sts tabell+5,temp
                ldi temp,0x5f
                sts tabell+6,temp
                ldi temp,0x70
                sts tabell+7,temp
                ldi temp,0x7f
                sts tabell+8,temp
                ldi temp,0x7b
                sts tabell+9,temp
                ldi temp,0xff
                out ddrb,temp
                out ddrc,temp
                clr sek1
                clr sek10
vistid:  mov tall,sek1
                rcall seg7
                out portb,kode
                mov tall,sek10
                rcall seg7
                out portc,kode
                ldi ventetid,1
                rcall vent

```

```

        inc sek1
        cpi sek1,10
        brne slutt1
        clr sek1
        inc sek10
        cpi sek10,6
        brne slutt2
        clr sek10
slutt2:
slutt1:
        rjmp vistid

seg7:
        ldi xl,low(tabell)
        ldi xh,high(tabell)
        clr r22
        add xl,tall
        adc xh,r22
        ld kode,x
        ret
vent:    ret    ; lager ikke kode for vent

.dseg
.org 0x100
tabell: .byte 10

```

;Alternativ B: Tabellen ligger i FLASH-minnet

```

.nolist
.include "m128def.inc"
.list
.def sek1=r20
.def sek10=r21
.def kode=r17
.def tall=r16
.def ventetid=r18
.def temp=r19
.org 0
        rjmp start

.org 0x46
start:   ldi temp,low(ramend)
        out spl,temp
        ldi temp,high(ramend)
        out sph,temp
        ldi temp,0xff
        out ddrb,temp
        out ddrc,temp
        clr sek1

```

DAT103

```
vistid:    clr sek10
           mov tall,sek1
           rcall seg7
           out portb,kode
           mov tall,sek10
           rcall seg7
           out portc,kode
           ldi ventetid,1
           rcall vent
           inc sek1
           cpi sek1,10
           brne slutt1
           clr sek1
           inc sek10
           cpi sek10,6
           brne slutt2
           clr sek10

slutt2:
slutt1:    rjmp vistid

           seg7:
           ldi zl,low(tabell*2)
           ldi zh,high(tabell*2)
           clr r22
           add zl,r16
           adc zh,r22
           lpm r17,z
           ret

vent:      ret           ;lager ikke kode for vent

Tabell:    .db 0x7e,0x30,0x6d,0x79,0x33,0x5b,0x5f,0x70,0x7f,0x7b
```