
Uppgift 1

Fråga 1: Vad blir flaggorna utsatta till nu?

N	Z	C
1	0	0

Fråga 2: Vad blir flaggorna utsatta till nu?

N	Z	C
0	1	1

Fråga 3: Vad blir flaggorna utsatta till om R0 har följande värden?

	R0 < 9	R0 = 9	R0 > 9
N	1	0	0
Z	0	1	0

Uppgift 2

Programmet ska med hjälp av BCD kod räkna upp till 99:

```
NAME    main
PUBLIC  main
SECTION .text : CODE (2)
THUMB

main
    MOV R0, #0 ; (R0) <-- 0
    MOV R1, #0 ; (R1) <-- 0
LOOP   CMP R1, #9 ; jämför (R1) med 9
       BNE ADD0 ; olika --> hoppa till ADD0
       ADD R0, R0, #7 ; uppdaterar först: (R0) <-- (R0)+7
       MOV R1, #0 ; (R1) <-- 0
       B ADD1 ; hoppa till ADD1
ADD0   ADD R0, R0, #1 ; (R0) <-- (R0) +1
       ADD R1, R1, #1 ; (R1) <-- (R1) +1
ADD1   CMP R0, #0x99 ; jämför (R0) med 99
       BNE LOOP ; olika --> hoppa till LOOP, upprepa
STOP   B STOP ; infinite lopp
END
```

Hur fungerar programmet?

R0 och R1 nollställs. Därefter inom en Loop jämför programmet R1 med 9. Om det går igenom nollställs R1 och R0 ökas med 7 i vårt fall. Om det inte går igenom skickas man till ADD0 som adderar på 1 på R0 och R1. Programmet jämför sedan R0 med 0x99 och ser om R0 är 99. Stämmer det avslutas programmet, om inte går programmet till början på Loopen och kollar om R1 = 9 och håller på så fram tills att R0 = 99.

Uppgift 3

Programmet ska hitta det minsta respektive största värdet i en vector (array):

```
NAME    main
PUBLIC  main
SECTION .text : CODE (2)
THUMB

main
    LDR R0, =vector ; (R0) <-- vector
    LDR R1, [R0], #4 ;läser varje 4 bit
    MOV R7, R1 ; (R7)<-- R1
    MOV R6, R1 ; (R6)<-- R1
    MOV R2, #0
LOOP    CMP R6, R1 ; jämför (R1) med 9
        BGT SET0 ; lägre --> hoppa till SET0
        BLE NEXT ; lägre --> hoppa till SET1
SET0    MOV R6, R1 ;set the value
NEXT    CMP R7, R1 ;compare R7 R1
        BLT SET1 ; större --> hoppa till SET1
        BGE NEXT2 ; jump o next NEXT2
SET1    MOV R7, R1 ; set th value
NEXT2   LDR R1, [R0], #4 ;jump to next value
        ADD R2, R2, #1
        CMP R2, #20 ; compare if the value is empty
        BNE LOOP ; olika --> hoppa till LOOP, upprepa
STOP    B STOP ; infinite lopp
ALIGNRO 2
DATA
vector
    DC32 14, 25, 2, 27, 3
    DC32 22, 13, 4, 24, 6
    DC32 26, 18, 8, 15, 9
    DC32 28, 10, 7, 17, 5
END
```

Hur fungerar programmet?

R0 laddas in med vectorn och R1 tilldelas så att den räknar varje 4bit. R6 som är det minsta värdet och R7 som är det största värdet får värdet av R1 i sig som startvärde. R2 får värdet 0. Loopen startar med att jämföra R6 med R1. ÄR R6 lägre än R1 hoppar programmet till SET0. Där får R6 (minsta nuvarande värdet) värdet av R1. Därefter följer SET0 samma procedur som om R6 är större än R1, programmet går till NEXT. Där jämförs R7 med R1, är det större hoppar den till SET1 och R7 (största nuvarande värdet) får R1 värdet. ÄR R7 mindre hoppar programmet till NEXT2 där R1 hoppar till nästa värde i vectorn. R2 fungerar som en counter i detta program. Varje gång Loopen gått igenom en gång adderas 1 till R2. R2 jämförs med 20 (antal element i vectorn), om det går igenom går programmet in

i en infinitivloop, om inte hoppar programmet till Loop och fortsätter till R2 är 20. Då har programmet fått u det största och minsta värdet i R6 respektive R7.