

**Navodila:**

Napišite zaporedje ukazov, ki vrednost enobajtne spremenljivke STEV1 prepiše v enobajtno spremenljivko STEV2. Nalogo rešite z neposrednim naslavljanjem (zaporedje lahko vsebuje le 2 ukaza v zbirniku za 68HC11) in indeksnim (baznim) naslavljanjem (zaporedje lahko vsebuje le 3 ukaze v zbirniku za 68HC11). Spremenljivki STEV1 in STEV2 naj bosta na naslovih od \$2000 dalje. Ukazi naj se nahajajo od naslova \$E000 naprej (ROM pomnilnik). Katere registre lahko uporabite?

**program01.asm**

```

ORG      $2000
STEV1    FCB      $5
STEV2    FCB      $0

ORG      $E000
START
LDA      STEV1
STA      STEV2
ZANKA   JMP      ZANKA

ORG      $FFFF
FDB      START

```

**Navodila:**

Napišite zaporedje ukazov, ki vrednost dvobajtne spremenljivke STEV1 prepiše v dvobajtno spremenljivko STEV2. Nalogo rešite z neposrednim naslavljanjem (zaporedje lahko vsebuje le 2 ukaza v zbirniku za 68HC11) in indeksnim (baznim) naslavljanjem (zaporedje lahko vsebuje le 3 ukaze v zbirniku za 68HC11). Koliko različnih zaporedij z 2 ukazoma (neposredno naslavljanje) in koliko različni zaporedij s 3 ukazi (indeksno (bazno) naslavljanje) lahko napišete v zbirniku za 68HC11? Katera so? Spremenljivki STEV1 in STEV2 naj bosta na naslovih od \$2000 naprej.

**program02.asm**

```

ORG      $2000
STEV1    FDB      $11A
STEV2    FDB      $0

ORG      $E000
START
LDD      STEV1
STD      STEV2
ZANKA   JMP      ZANKA

ORG      $FFFF
FDB      START

-----
INDEKSNO
-----
LDX      #STEV1
LDD      0,X
STD      1,X

```

**Navodila:**

Napišite zaporedje ukazov v zbirniku za 68HC11, ki zamenja vrednosti enobajtnih spremenljivk STEV1 in STEV2. Naredite dve različici zaporedij z različnimi načini naslavljanja:

- Neposredno naslavljanie (4 ukazi),
  - Indeksno (bazno) naslavljanje (5 ukazov).
- Spremenljivki STEV1 in STEV2 naj bosta na naslovih od \$2000 naprej. Rešite še primer, ko sta spremenljivki STEV1 in STEV2 16 bitni. Število ukazov naj ostane enako. Ukazi v zbirniku se morajo nahajati od naslova \$E000 dalje (ROM pomnilnik).

**program03.asm**

```

ORG      $2000
STEV1    FDB      $11A
STEV2    FDB      $0

ORG      $E000

```

```

START      LDD      STEV1
          LDX      STEV2
          STD      STEV2
          STX      STEV1
ZANKA     JMP      ZANKA

          ORG      $FFFFE
          FDB      START

-----
INDEKSNO
-----
LDY      #STEV1
LDX      0,Y
LDD      1,Y
STX      1,Y
STD      0,Y

```

## Navodila:

Na naslovu \$2000 rezervirajte prostor za 5 bajtno spremenljivko z oznako TABELA. Nato napišite zaporedje ukazov v zbirniku za 68HC11, ki v vse baje spremenljivke TABELA zapiše vrednost 0. Nalogo rešite z indeksnim in neposrednim naslavljanjem! Uporabite le ukaze za nalaganje vrednosti v registre in ukaze za shranjevanje vrednosti registrov v pomnilnik. Ukazi v zbirniku se morajo nahajati od naslova \$E000 dalje (ROM pomnilnik).

```

program04.asm
          ORG      $2000
TABELA    RMB      5

          ORG      $E000
START
          LDA      #0
          STA      TABELA
          STA      TABELA+1
          STA      TABELA+2
          STA      TABELA+3
          STA      TABELA+4
ZANKA     JMP      ZANKA

          ORG      $FFFFE
          FDB      START
-----
INDEKSNO
-----
          LDA      #0
          LDX      #TABELA
          STA      0,X
          STA      1,X
          STA      2,X
          STA      3,X
          STA      4,X

```

## Navodila:

Na naslovu \$2000 rezervirajte prostor za 10 bajtno spremenljivko z oznako TABELA, ki ima začetne vrednosti števila od 1 do 10. Nato napišite zaporedje ukazov v zbirniku za 68HC11, ki zamenja vrstni red elementov spremenljivke TABELA tako, da so po zamenjavi vrstnega reda števila razporejena v obratnem vrstnem redu kot na začetku. Nalogo rešite z indeksnim in neposrednim naslavljanjem! Uporabite le ukaze za nalaganje vrednosti v registre in ukaze za shranjevanje vrednosti registrov v pomnilnik. Ukazi v zbirniku se morajo nahajati od naslova \$E000 dalje (ROM pomnilnik).

```

program05.asm
          ORG      $2000
TABELA    FDB      $102,$304,$506,$708,$90A

          ORG      $E000

```

```

START
    LDA    TABELA
    LDB    TABELA+9
    STA    TABELA+9
    STB    TABELA

    LDA    TABELA+1
    LDB    TABELA+8
    STA    TABELA+8
    STB    TABELA+1

    LDA    TABELA+2
    LDB    TABELA+7
    STA    TABELA+7
    STB    TABELA+2

    LDA    TABELA+3
    LDB    TABELA+6
    STA    TABELA+6
    STB    TABELA+3

    LDA    TABELA+4
    LDB    TABELA+5
    STA    TABELA+5
    STB    TABELA+4

ZANKA    JMP    ZANKA

ORG    $FFFFE
FDB    START
-----
INDEKSNO
-----
    LDX    #TABELA
    LDA    0,X
    LDB    9,X
    STA    9,X
    STB    0,X
    LDA    1,X
    LDB    8,X
    STA    8,X
    STB    1,X
    LDA    2,X
    LDB    7,X
    STA    7,X
    STB    2,X
    LDA    3,X
    LDB    6,X
    STA    6,X
    STB    3,X
    LDA    4,X
    LDB    5,X
    STA    5,X
    STB    4,X

```

Pridružil/a: 03 Julij 2003

Lokacija: Slovenija

Objave: 4042

Objavljeno: 02 Januar 2005 ob 13:45 | Shranjen IP



Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki sešteje enobajtni spremenljivki STEV1 in STEV2, rezultat pa zapiše v enobajtno spremenljivko REZ. Napišite 3 različice zaporedja ukazov v 2 načinih naslavljanja (neposredno in indeksno(bazno)): ena naj uporablja le akumulator A, ena le akumulator B, ena pa oba akumulatorja. Spremenljivke STEV1, STEV2 in REZ naj bodo na naslovih od \$2000 naprej. Nalogu ponovite še na primeru, če spremenljivko STEV2 odštejemo od spremenljivke STEV1.

## program06.asm

```

        ORG    $2000
STEV1   FCB    $5
STEV2   FCB    $6
REZ     FCB    $0

        ORG    $E000
START   LDA    STEV1
        ADDA   STEV2
        STA    REZ ; prvi nacin

        LDB    STEV1
        ADDB   STEV2
        STB    REZ ; drugi nacin

        LDA    STEV1
        LDB    STEV2
        ABA
        STA    REZ ; drugi nacin

ZANKA   JMP    ZANKA

        ORG    $FFFE
        FDB    START

-----
INDEKSNO
-----
        LDX    #STEV1
        LDA    0,X
        ADDA   1,X
        STA    2,X

        LDX    #STEV1
        LDB    0,X
        ADDB   1,X
        STB    2,X

        LDX    #STEV1
        LDA    0,X
        LDB    1,X
        ABA
        STA    2,X
ali      ORG    $2000
STEV1   FCB    $6
STEV2   FCB    $5
REZ     FCB    $0

        ORG    $E000
START   LDA    STEV2
        NEGA
        STA    STEV2
; negirali smo STEV1 da jo odstejemo
        LDA    STEV1
        ADDA   STEV2
        STA    REZ ; prvi nacin

        LDB    STEV1
        ADDB   STEV2
        STB    REZ ; drugi nacin

        LDA    STEV1
        LDB    STEV2
        ABA
        STA    REZ ; drugi nacin

```

ZANKA	JMP	ZANKA
	ORG	\$FFFF
	FDB	START

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki sešteje dvobajtni spremenljivki STEV1 in STEV2, rezultat pa zapiše v dvobajtno spremenljivko REZ. Uporabite ukaze 68HC11 za 16 bitno seštevanje. Rešite nalogu z neposrednim in indeksnim (baznim) načinom naslavljanja. Spremenljivke STEV1, STEV2 in REZ naj bodo na naslovih od \$2000 naprej.

program07.asm

```

        ORG    $2000
STEV1   FDB    $100
STEV2   FDB    $150
REZ     FDB    $0

        ORG    $E000
START   LDD    STEV1
        ADDD   STEV2
        STD    REZ
ZANKA   JMP    ZANKA

        ORG    $FFFF
FDB    START

-----
INDEKSNO
-----

        LDX    #STEV1
LDD    0,X
ADDD   2,X
STD    4,X

```

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki sešteje 32-bitni spremenljivki STEV1 in STEV2, rezultat pa zapiše v 32-bitno spremenljivko REZ. Uporabite ukaze 68HC11 za 8 bitno seštevanje in pravilno upoštevajte prenos. Nalogo rešite z neposrednim in indeksnim (baznim) načinom naslavljanja. Spremenljivke STEV1, STEV2 in REZ naj bodo na naslovih od \$2000 naprej. Nalogo rešite brez uporabe zanke.

Napišite tudi zaporedje ukazov za 32 bitno odštevanje.

program08.asm

```

-----
INDEKSNO
-----

        ORG    $2000
STEV1   FCB    $01,$02,$03,$04
STEV2   FCB    $05,$06,$07,$08
REZ     RMB    4
        ORG    $E000
START   LDX    #STEV1
        LDAA   3,X
        ADDA   7,X
        STAA   11,X

        LDAA   2,X
        ADCA   6,X
        STAA   10,X

        LDAA   1,X
        ADCA   5,X
        STAA   9,X

        LDAA   0,X
        ADCA   4,X

```

STAA	8,X
------	-----

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki vrednost osembitne spremenljivke STEV1 najprej zmanjša za 1, nato jo naloži v akumulator A, jo poveča za 1, prepiše v akumulator B, zmanjša vrednost v B za 1 in dobljeno vrednost zapiše nazaj v spremenljivko STEV1, ki jo na koncu poveča za 1. Spremenljivka STEV1 naj bo na naslovu \$2000.

program09.asm

STEV1	ORG	\$2000
	FCB	\$5
	ORG	\$E000
	DEC	STEV1
	LDAA	#STEV1
	INC	STEV1
	LDAB	STEV1
	DEC B	
	STAB	STEV1
	INC	STEV1

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki vrednost šestnajstbitne spremenljivke STEV1 poveča za 1, vrednost šestnajstbitne spremenljivke STEV2 pa zmanjša za 1. Spremenljivki naj bosta na naslovih od \$2000 naprej. Nalogo rešite na dva načina: z uporabo registrov (4 izvedbe s 6 ukazi) in brez uporabe registrov\* (6 ukazov). Pozor: ena rešitev z uporabo registrov je "nevarna" in se ne uporablja. Katera?

program10a.asm

a)	STEV1	ORG	\$2000	
	STEV2	FDB	\$AAAA	
		FDB	\$1111	
	START	ORG	\$E000	
		LDI	STEV1	
		ADD D	#1	
		STD	STEV1	
		LDI	STEV2	
		SUB D	#1	
		STD	STEV2	
b)	STEV1	ORG	\$2000	
	STEV2	FDB	\$AAAA	
		FDB	\$1111	
	START	ORG	\$E000	
		LDI	STEV1	
		INX		
		STX	STEV1	
		LDX	STEV2	
		DEX		
		STX	STEV2	
c)	(nevarna, ker se igramo s kazalcem na stack, in potem zgubimo lokacijo kjer smo nazadnje brali/pisali s stacka)	STEV1	ORG	\$2000
		STEV2	FDB	\$AAAA
			FDB	\$1111
		START	ORG	\$E000
			LDS	STEV1
		INS		
		STS	STEV1	
		LDS	STEV2	
		DES		
		STS	STEV2	
d)	STEV1	ORG	\$2000	
	STEV2	FDB	\$AAFE	
		FDB	\$1101	
		ORG	\$E000	
	START	INC	STEV1+1	
		BNE	NAPREJ	

	INC	STEV1
NAPREJ	TST	STEV2+1
	BNE	KONEC
	DEC	STEV2
KONEC	DEC	STEV2+1
K	BRA	K
	ORG	\$FFFF
	FDB	START

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki v akumulator A najprej naloži vrednost 255. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program prišteje akumulatorju A vrednost 1. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program še enkrat prišteje akumulatorju A vrednost 1. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj?

Kolikšna je končna vrednost v akumulatorju A?

## program11.asm

```
LDAA #255
Z se postavi na 0, ker $FF ni vrednost $00
C ostane, ker pri nalaganju vrednosti ne pride do prenosa
V se postavi na 0, ker ni preliva
N se postavi na 1, ker je $FF ubistvu vrednost -1
    LDAA #255
    INCA
Z se postavi na 1, ker $FF + $1 == $00
C se pri INCA ne spremeni, pri ADDA #1 pa se C spremeni v 1 zaradi prenosa
V ostane na 0, ker ni preliva
N se postavi na 0, ker $00 ni negativno stevilo
    LDAA #255
    INCA
    INCA
Z se postavi na 0, ker A nima vec vrednosti $00
C se pri INCA ne spremeni, pri ADDA #1 pa tudi ne, ker $00 + $01 == $01 in ni prenosa
V ostane na 0, ker ni preliva
N ostane 0, ker $01 tudi ni negativno stevilo
```

Vrednost v akumulatorju A je \$01.

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki v akumulator A najprej naloži vrednost 127. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program prišteje akumulatorju A vrednost 129. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Kolikšna je končna vrednost v akumulatorju A?

## program13.asm

```
Napišite zaporedje ukazov v zbirniku za 68HC11, ki v akumulator A najprej naloži vrednost 127. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program prišteje akumulatorju A vrednost 129. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Kolikšna je končna vrednost v akumulatorju A?
```

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki v akumulator A najprej naloži vrednost 1. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program odšteje od akumulatorja A vrednost 1. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program še enkrat odšteje vrednost 1 od akumulatorja A. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Kolikšna je končna vrednost v akumulatorju A?

## program21.asm

```
LDAA #
Z se postavi na 0, ker $01 != $00
C ostane, ker pri nalaganju vrednosti ne pride do prenosa
V se postavi na 0, ker ni preliva
N se postavi na 0, ker $01 ni negativno stevilo
    LDAA #
    DECA
Z se postavi na 1, ker $01 - $01 == $00
```

C se pri INCA ne spremeni, pri SUBA #1 pa tudi ne, ker \$01 - \$01 == \$00 in se ni prenosa  
 V ostane na 0, ker ni preliva  
 N ostane 0, ker \$00 ni negativno stevilo  
 LDAA #1  
 DECA  
 DECA  
 Z se postavi na 0, ker \$FF != \$00  
 C se pri INCA ne spremeni, pri SUBA #1 pa se postavi na 1  
 V ostane na 0, ker ni preliva  
 N se spostavi na 1, ker je \$00 - \$01 == \$FF in \$FF je -1  
 Vrednost v akumulatorju A je \$FF

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki v akumulator A najprej naloži vrednost 0. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program naloži v akumulator B vrednost 255. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Nato naj program odšteje akumulator B od akumulatorja A. Kakšne vrednosti imajo zastavice Z, C, V in N po tem ukazu? Zakaj? Kolikšna je končna vrednost v akumulatorju A?

## program14.asm

```

LDAA #0
Z = 1 ; ker je #0 enaka 0
V = 0 ; se avtomatsko nastavi vedno ob ukazu LDAA
N = 0 ; ker je #0 v pozitivnem obmocju
C = ? ; ker LDAA ne spremeni carry flaga

LDAB #255
Z = 0 ; ker #255 ni enaka 0
V = 0 ; se avtomatsko nastavi vedno ob ukazu LDAA
N = 1 ; ker je #255 == -1
C = ? ; ker LDAA ne spremeni carry flaga

SBA
Z = 0; ker #1 ni enaka 0
V = 0; ker ne pride do preliva (SUB&-1=+1)
N = 0; ker je #1 v pozitivnem obmocju
C = 1; ker od 0 odstevamo $FF, to pomeni da pride do prenosa in se skoraj cel krog do 1

```

Koncna vrednost v A = #1

## program14.asm

```

org    $FFFE      ;*
fdb   start       ;*
; -----
org    $2000
vrednost fcb 0
vrednost2 fcb 255

org    $e000
start
ldaa  vrednost    ; ali druga možnost #127
ldab  vrednost2   ; ali druga možnost #129
sba

; -----
konec bra  konec   ;* relativno naslavljanje
end
;
```

## Navodila:

Kateri od naslednjih pogojnih skokov se bo izvršil, če je stanje zastavic N=1, C=1, Z=0 in V=0?

- a) BCC
- b) BNE
- c) BGE
- d) BLS
- e) BMI

- f) BCS  
g) BLT

**program16.asm**

```
BNE, BLS, BMI, BCS, BLT
samo BCC in BGE ne preskocita.
BNE: Ce Z=0 (razlicno od 0) potem preskoci
BLS: Ce vsaj eden od C=1 ali Z=1 potem preskoci
BMI: Ce N=1 (negativno) potem preskoci
BCS: Ce C=1 (prenos) potem preskoci
BLT: Ce samo eden od N=1 ali V=1 potem preskoci

BCC: Preskocilo bi, ce bi bil C=0
BGE: Preskocilo bi, ce bi bila OBA N,V=1 ali OBA N,V=0
```

Navodila:

Kateri od naslednjih programov se vedno vrtijo v zanki L1?

- a)  
L1 LDAA #127  
BNE L1
- b)  
L1 LDAA #127  
BEQ L1
- c)  
L1 LDAA #0  
BEQ L1
- d)  
L1 LDAA #1  
CMPA #25  
BMI L1
- e\*)  
L1 LDAA 1  
BPL L1

**program17.asm**

```
a, ker #127 != #0, zato preskoci
c, ker #0 == #0, zato preskoci
d, ker bo #1 vedno manjse od #25. to vedno dela SUBA #25 in dobimo C,N=1, V,Z=0,
nato spet LDAA #1. in ker je N=1, preskoci

b ne, ker #127 != #0, zato ne preskoci

e 50%:50%, enkrat bo, enkrat ne bo, odvisno od tega kaj je na lokaciji 1 v RAMu. Ce je N=0 (127 >= mem[1] >= 0) potem preskoci
```

Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki najprej zapiše dvojiški komplement enobajtne spremenljivke STEV1 v spremenljivko STEV2, nato pa eniški komplement STEV1 zapiše v STEV3. Kakšna je končna vrednost spremenljivk STEV2, STEV3, če je začetna vrednost spremenljivke STEV1 enaka: 0, 10, \$FE, -1, 32, 128, %01011010? Spremenljivke STEV1, STEV2 in STEV3 naj bodo na naslovih od \$2000 naprej. Napišite zaporedje ukazov za neposredno in indeksno (bazno) naslavljanje.

**program18.asm**

Koda za primer STEV1=0:

Neposredno:

```
ORG $2000
STEV1 FCB 0
STEV2 RMB 1
STEV3 RMB 1
ORG $E000
START LDAA STEV1
NEGA
STA STEV2
LDAA STEV1
COMA
STA STEV3
KONEC BRA KONEC
ORG $FFFF
```

## FDB START

```

Indeksno:
    ORG $2000
STEV1   FCB 0
STEV2   RMB 1
STEV3   RMB 1
        ORG $E000
START   LDX #STEV1
        LDAA 0,X
        NEGA
        STAA 1,X
        LDAA 0,X
        COMA
        STAA 2,X
KONEC   BRA KONEC
        ORG $FFFE
        FDB START

```

## Tabela vrednosti:

STEV1	STEV2	STEV3
0	0	\$FF
10	\$F6	\$F5
\$FE	02	01
\$FF	1	0
32	\$E0	\$DF
128	\$80	\$7F
%01011010	\$A6	\$A5

## Navodila:

Napišite zaporedje ukazov v zbirniku za 68HC11, ki zapiše dvojiški komplement dvobajtne spremenljivke STEV1 v spremenljivko STEV2. Spremenljivki STEV1 in STEV2 naj bosta na naslovih od \$2000 naprej. Napišite zaporedje ukazov za neposredno in indeksno (bazno) naslavljvanje. \*Nalogo rešite še za primer 32-bitnih spremenljivk.

program19.asm

```

Neposredno:
    ORG    $2000
STEV1  RMB   2
STEV2  RMB   2
        ORG    $E000
START   LDAA   STEV1+1
        NEGA
        STAA   STEV2+1
        LDAA   STEV1
        INCA
        NEGA
        STAA   STEV2
KONEC   BRA    KONEC
        ORG    $FFFE
        FDB    START

```

## Indeksno:

```

        ORG    $2000
STEV1  RMB   2
STEV2  RMB   2
        ORG    $E000
START   LDX    #STEV1
        LDAA   1,X
        NEGA
        STAA   3,X
        LDAA   0,X
        INCA
        NEGA
        STAA   2,X
KONEC   BRA    KONEC
        ORG    $FFFE
        FDB    START

```

32-bitne spremenljivke:

```

ORG    $2000
STEV1  RMB    4
STEV2  RMB    4
        ORG    $E000
START   LDX    #STEV1
        LDAA   3,X
        NEGA
        STAA   7,X
        LDAA   2,X
        INCA
        NEGA
        STAA   6,X
        LDAA   1,X
        INCA
        NEGA
        STAA   5,X
        LDAA   0,X
        INCA
        NEGA
        STAA   4,X
KONEC  BRA    KONEC
        ORG    $FFFE
        FDB    START

```

## Navodila:

Zapišite zaporedje ukazov za 68HC11, ki zamenja vrednosti kumulatorjev A in B. Pomožna spremenljivka naj bo na naslovu \$2000. Naredite dve različici s 3 ukazi brez uporabe sklada.

## program20.asm

```

1.:
        ORG    $2000
STEV1  RMB    2
        ORG    $E000
START   STD    STEV1
        LDAA   STEV1+1
        LDAB   STEV1
KONEC  BRA    KONEC
        ORG    $FFFE
        FDB    START

2.:
        ORG    $2000
STEV1  RMB    1
        ORG    $E000
START   STAA   STEV1
        TBA
        LDAB   STEV1
KONEC  BRA    KONEC
        ORG    $FFFE
        FDB    START

```

## Navodila:

Zapišite zaporedje ukazov za 68HC11, ki zamenja vsebine registrov X in Y na 2 različna načina: preko pomnilnika in z zamenjavo vrednosti registrov. Pri zamenjavi preko pomnilnika naj bodo pomožne spremenljivke na naslovu \$2000.

## program21.asm

```

1.:
        ORG    $E000
START   XGDX
        XGDY
        XGDX
KONEC  BRA    KONEC
        ORG    $FFFE
        FDB    START

2.:
        ORG    $2000

```

```

STEV1 RMB 2
STEV2 RMB 2
    ORG $E000
START STX STEV1
    STY STEV2
    LDX STEV2
    LDY STEV1
KONEC BRA KONEC
    ORG $FFFE
    FDB START

```

## Navodila:

Napišite zaporedje ukazov za 68HC11, ki v akumulator A naloži rednost 10, nato pa akumulator stalno zmanjšuje za 1, dokler vrednost ni enaka 0 (programska zanka).

## program22.asm

```

ORG $E000
START LDAA #10
PREVERI DECA
    BNE PREVERI
KONEC BRA KONEC
    ORG $FFFE
    FDB START

```

## program22.asm

```

org $ffe
fdb start
; -----
org $2000 ; naslov
start

ldaa #10

loop suba #1
    cmpa #0
    bne loop
; -----
konec
bra konec
end

```

## Navodila:

Napišite zaporedje ukazov za 68HC11, ki postavi vse bajte spremenljivke (polja) TABELA na naslovu \$2000 na 0. Dolžina polja TABELA je 7 bajtov. Izdelajte dve različici. Pri eni uporabite rešitev iz naloge 22 (števec ponovitev), pri drugi pa uporaba števca ponovitev ni dovoljena.

Program pri prvi različici ne sme uporabiti akumulatorja B,  
pri drugi pa ne A in ne B. Pojasnite, zakaj pri uporabi števca uporabimo tehniko štetja navzdol!

## program23.asm

```

ORG $2000
TABELA RMB 7
ZAC RMB 2
    ORG $E000
    LDX #TABELA
    LDAA #7
    STAA ZAC
    CLRA
PREVERI ADDA 0,X
    STAA ZAC+1
    LDAA ZAC
    TST
    BNE PREVERI
    ORG $FFFE
    FDB $E000

```

## program23.asm

```

org $ffe
fdb start
; -----
org $2000 ; naslov

```

```

tabela    rmb    7
;tabelaK   fcb    0
tabelaK           ; boljša možnost

          org    $efff
start      idx    #tabela

zanka     clr    0,x    ; vpisovanje, brisanje 0
          inx    ; povečamo indeks
          cmpx   #tabelaK
          bne    zanka
; -----
konec      bra    konec
end

```

## program23.asm

```

          org    $ffe
          fdb    start
;

          org    $2000    ; naslov

tabela    rmb    7
stevec    fcb    7

          org    $efff
start      ;
          ldab   #0      ; z uporabo akumulatorja B
          ldaa   stevec
          idx    #tabela

zanka     stab   0,x
; ko se premikamo po tabeli je potrebno uporabiti indeksno nastavljanje
zanka     clr    0,x    ; vpisovanje, brisanje 0
          inx    ; povečamo indeks
          deca   ; odštejemo b
          bne    zanka
;
konec      bra    konec
end

```

## Navodila:

Napišite zaporedje ukazov za 68HC11, ki v akumulator A naloži vrednost 5 in pobriše akumulator B. Program naj nato akumulator A stalno zmanjšuje za 1, dokler vrednost ni enaka 0, akumulatorju B pa naj prišteva 55. Kolikšna je končna vrednost v akumulatorjih A in B? (dolžina zaporedja je 5 ukazov v zbirniku).

## program24.asm

```

          ORG    $E000
          LDAA   #5
          CLRB
NAZAJ     ADDB   #55
          DECA
          BNE    NAZAJ
KONEC    BRA    KONEC
          ORG    $FFFE
          FDB    $E000

;A=$00
;B=$13

```

## program24.asm

```

          org    $ffe
          fdb    start
;

          org    $2000    ; naslov

```

```

org    $efff

start
    ldaa   #5
    clrb

loop  suba   #1
      addb   #55
      cmpa   #0
      bne    loop

; Vrednost v A akumulatorju = 0;
; Vrednost v B akumulatorju = 19;
; -----
konec
    bra    konec
end

```

## Navodila:

Rešite nalog 24 še tako, da za števec namesto akumulatorja A uporabite spremenljivko STEVEC na naslovu \$2000, akumulatorju B pa prištevajte vrednost spremenljivke KON1, ki naj bo na naslovu \$2010. Nalogo rešite z uporabo neposrednega (a) in indeksnega (baznega) naslavljanja (b).

## program25a.asm

```

a)
        ORG      $2000
STEVEC  FCB      5
        ORG      $2010
KON1    FCB      55
        ORG      $E000
        CLRBL
NAZAJ   ADDB     KON1
        DEC      STEVEC
        BNE      NAZAJ
KONEC   BRA      KONEC
        ORG      $FFFFE
        FDB      $E000

```

## program25b.asm

```

b)
        ORG      $2000
STEVEC  FCB      5
        ORG      $2010
KON1    FCB      55
        ORG      $E000
        CLRBL
        LDX      #STEVEC
NAZAJ   ADDB     $10,X
        DEC      STEVEC
        BNE      NAZAJ
KONEC   BRA      KONEC
        ORG      $FFFFE
        FDB      $E000

```

## Navodila:

. Napišite zaporedje ukazov, ki v zanki 100 krat izvede ukaz NOP. Nalogo rešite tudi za primer, ko se ukaz NOP izvede 1000 krat in 100000 krat(\*). Nalogo rešite:

- a) z uporabo le ene zanke;
- b) \*z uporabo vgnezdjenih zank z 8-bitnimi števci.

Za vsak primer naredite 2 različici: ena naj uporablja le števce v registrih, druga pa le števce v pomnilniku!

## program26a.asm

100x ena zanka pomnilnik:
---------------------------

	ORG	\$2000
STEVEC	FCB	100
	ORG	\$E000
ZANKA	NOP	
	DEC	STEVEC
	BNE	ZANKA
KONEC	BRA	KONEC
	ORG	\$FFFF
	FDB	\$E000

100x ena zanka register:

	ORG	\$2000
	ORG	\$E000
	LDAA	#100
ZANKA	NOP	
	DECA	
	BNE	ZANKA
KONEC	BRA	KONEC
	ORG	\$FFFF
	FDB	\$E000

## program26b.asm

1000x dve zanki pomnilnik:

	ORG	\$2000
STEVEC	FCB	\$3, \$E8
	ORG	\$E000
ZANKA	NOP	
	DEC	STEVEC+1
	BNE	ZANKA
	DEC	STEVEC
	BPL	ZANKA
KONEC	BRA	KONEC
	ORG	\$FFFF
	FDB	\$E000

1000x ena zanka register:

	ORG	\$E000
	LDX	#1000
ZANKA	NOP	
	DEX	
	BNE	ZANKA
KONEC	BRA	KONEC
	ORG	\$FFFF
	FDB	\$E000

## program26c.asm

100000x 5 skokov pomnilnik:

	ORG	\$2000
STEVEC	FCB	\$1, \$86, \$A0
	ORG	\$E000
ZANKA	NOP	
	DEC	STEVEC+2
	BNE	ZANKA
	DEC	STEVEC+1
	BPL	ZANKA
	DEC	STEVEC
	BPL	ZANKA
	INC	STEVEC+1
	BEQ	KONEC
	DEC	STEVEC+1
	BRA	ZANKA
KONEC	BRA	KONEC
	ORG	\$FFFF
	FDB	\$E000

100000x dve zanki register:

	ORG	\$E000
	LDAA	#\$1
	LDX	#\$86A0 ; \$186A0 == 100000
ZANKA	NOP	

	DEX	
	BNE	ZANKA
	DECA	
	BPL	ZANKA
KONEC	BRA	KONEC
	ORG	\$FFFF
	FDB	\$E000

Navodila:

Napišite zaporedje ukazov za 68HC11, ki enobajtno spremenljivko STEV1 pomnoži s 4, rezultat pa shrani v enobajtno spremenljivko REZ. Kolikšen je rezultat če za STEV1 vzamemo 0, 1, 5, 59 in 71? Ukaz za množenje ni dovoljen. Namig: množenje z  $2^n$  je pomik v levo za  $n$  mest. Spremenljivki naj se nahajata na zaporednih naslovih od \$2000 naprej. Rešite nalogu še tako, da imate na razpolago le spremenljivko STEV1, v kateri naj bo na koncu shranjen tudi rezultat. Pri tem ne smete uporabiti ne akumulatorja A ne B. Nalogo rešite z uporabo neposrednega in indeksnega (baznega) naslavljanja.

program28.asm

```

        ORG      $2000
STEV1   FCB      71
REZ     RMB      1
        ORG      $E000
        LDAA    STEV1
        ASLA
        ASLA
        STAA    REZ
KONEC   BRA     KONEC
        ORG      $FFFF
        FDB      $E000

Vnos    Rezultat
0       $0
1       $4
59      $EC
71      $1C, ker $1 pri 2. bajtu odreze (saj smo uporabili samo 8 bitov)

```

Indeksno:

```

        ORG      $2000
STEV1   FCB      71
REZ     RMB      1
        ORG      $E000
        LDX     #STEV1
        LDAA    0,X
        ASLA
        ASLA
        STAA    1,X
KONEC   BRA     KONEC
        ORG      $FFFF
        FDB      $E000

```

Samo STEV1:

```

        ORG      $2000
STEV1   FCB      71
REZ     RMB      1
        ORG      $E000
        ASL     STEV1
        ASL     STEV1
KONEC   BRA     KONEC
        ORG      $FFFF
        FDB      $E000

```

Navodila:

Napišite zaporedje ukazov za 68HC11, ki enobajtno nepredznačeno spremenljivko STEV1 deli z 8, rezultat (količnik in ostanek) pa shrani v spremenljivki KOL in REZ. Kolikšen je rezultat če za STEV1 vzamemo 0, 8, 15, 31, 129 in 250? Ukaz za deljenje ni dovoljen. Spremenljivke naj se nahajajo na zaporednih naslovih od \$2000 naprej. Rešite nalogu še za primer predznačenih števil. Pojasnite, zakaj so količniki pri deljenju lihih negativnih števil po absolutni vrednosti različni od količnikov pri deljenju lihih pozitivnih števil.

program29.asm

Nepredznačena stevila:

```

        ORG      $2000
STEV1   FCB      250
KOL     RMB      1
REZ     RMB      1
        ORG      $E000
        LDBA    STEV1
        TBA
        ASRB
        ASRB
        ASRB
        ANDB    #00011111
        STAB    KOL
        ASLB
        ASLB
        ASLB
        SBA
        STAA    REZ
KONEC   BRA     KONEC
        ORG      $FFFF
        FDB      $E000

STEV1   KOL      REZ
0       $0       $0
8       $1       $0
15      $1       $7
31      $3       $7
129     $10      $1
250     $1F      $2

```

Predznacena stevila:

...

#### Navodila:

Napišite program, ki 8-bitno spremenljivko S1 pomnoži s 3 (uporabite seštevanje: S1+S1+S1), nato pa od vsote odšteje konstanto K. Konstanta K naj ima vrednost 101101 (Bin), za spremenljivko S1 pa samo rezervirajte prostor. Na začetku programa v S1 naložite začetno vrednost 5A (Hex). Ob vsakem ukazu kot komentar napišite stanje zastavic (N,Z,V,C) in način naslavljanja, npr.:

ADDA #1 ;N=1, Z=0, V=1, C=0, takojšnje

Program naj se zacne na naslovu E000 (Hex), spremenljivke pa na naslovu 2000 (Hex).

Utemeljite, zakaj je stanje vseh zastavic po izvedbi zadnjega ukaza tako kot ste napisali (pod programom napišite komentar), npr.:

;N = 1, ker je v A število, ki je večje od 127 in je torej negativno (predznačeno število).

;Z = 0, ker rezultat ni enak 0.

;V = 1, ker smo številu 127 prišeli 1 in prekoračili obseg predznačenih števil

;C = 0, ker pri 127+1 ne prekoračimo obsega nepredznačenih števil

Kakšna je končna vrednost spremenljivke S1, če gre za nepredznačeno število?

#### primer\_programa.asm

; Ime in priimek, vpisna številka

```

        org  $ffff
        fdb  start
;
; -----
org  $2000      ; začetek spremenljivk
s1   rmb  1       ; ker je 8-bitna
k    equ  %01101    ; konstanto se vpiše z EQU, ker ne želimo rezervirati prostor

;
; -----
org  $e000 ; začetek programa
start
ldaa #5a ; NZVC=0000 - takojšnje
staa s1 ; NZVC=0000 - razširjeno neposredno (ker je naslov nad 2000)
adda s1 ; + ; NZVC=1010 - razširjeno neposredno
adda s1 ; + ; NZVC=0001 - razširjeno neposredno
suba #k ; - ; NZVC=1001 - takojšnje

```

```

staa s1      ; NZVC=1001 - razširjeno neposredno
; N=1 ker smo s staa shranili negativno število iz akumulatorja A
; Z=0 ker število, ki smo ga shranili ni nič
; V=0 ker ukaz staa v vedno postavi na 0
; C=1 = staa ga ne spreminja, v prejšnjem ukazu pa se je postavil na
; 1 ker smo od 14 odšteli 45

; Končna vrednost: E1
; N=1

; -----
konec      bra      konec ;           - relativno
end

```

Navodila:

**Prvo preverjanje znanja pri predmetu ORA1, ponedeljek, skupina A**

Imamo 3 osemtbitne spremenljivke STEV1, STEV2 in REZULTAT. Prvi dve, ki imata začetni vrednosti 01110000B in 28D, se nahajata od naslova 2000H dalje, zadnja, ki je brez začetne vrednosti, pa je na naslovu 2010H.

- Napišite program za 68HC11, ki osemtitno sešteje STEV1 in STEV2, nato pa od vsote odšteje konstanto DH. Končni rezultat shranite v spremenljivko REZULTAT. Program naj se nahaja od naslova \$E000 dalje.
- Kot komentar k vsaki vrstici napišite uporabljeni način naslavljanja.
- Kot komentar na koncu programa napišite kakšno je stanje zastavic N, Z, C in V po odštevanju. Zakaj?

program.asm

```

org    $ffe          ;
fdb   start          ;
; -----
org    $2000         ; neposredno
stev1  fcb  %01110000 ; neposredno
stev2  fcb  28        ; neposredno

rezultat org    $2010         ; neposredno
          rmb  1           ; neposredno

start      org    $e000         ; neposredno
          ldaa  stev1        ; neposredno
          ldab  stev2        ; neposredno
          aba   ; IHN...registrsko naslavljanje
          suba  #$d          ; takojšne
          staa  rezultat     ; neposredno
; tukaj lahko uporabimo ukaze, staa, sta, std,... rezultat je še vedno enak!
; -----
konec      bra   konec        ; relativno
end

; Stanje zastavic pred odštevanjem:
; N=1, Z=0, C=0, V=1

; Stanje zastavic po odštevanju:
; N=0, Z=0, C=0, V=1

```