PROBLEMA 2 (COSINUS TABELA * 2^14)

```
.include "m8535def.inc"
.dseg
NUMAR:      .byte       5
.cseg
RESET:
        rjmp    INIT
CAUTA:
        ldi R20,4
        mov ZL,R20
        clr ZH
        lsl ZL
        rol ZH
        subi ZL,low(-2*TAB_SEG)
        sbci ZH,high(-2*TAB_SEG)
        lpm R17,Z+
        lpm R18,Z
; folosesc 10biti pentru adresarea lui 1000 ... si inac 14 - total 24 = 3 Registrii R16:17:18
        clr R16

        lsr R18
        ror R16
        lsr R18
        ror R16
        mov R18,R16
        clr R16

        lsr R17
        ror R16
        lsr R17
        ror R16
        add R18,R17
        mov R17,R16
        clr R16
ret
; Tabel de conversie cifra-reprezentare pe 7 segmente
TAB_SEG:
        .dw     1000,999,998,997,242,992,989,985,980,975,970,963,956,949,941
;dw – define word, pt ca este pe 10 biti
INIT:
        ldi     R16,low(RAMEND)
        ldi     R17,high(RAMEND)
        out     SPL,R16
        out     SPH,R17

        rjmp    CAUTA
rjmp INIT
```

PROBLEMA 3 (PORTA SEMNAL 0-1)

```
.include "m8535def.inc"
.equ    VAL1  = 100 ;pt a defini 10ns pentru valoarea 1
.equ    VAL0  = 200;pt a defini 20ns pentru valoarea 0
.cseg
        rjmp    INIT
; Adresa de intrare in intreruperea de comparatie canal A
        .org    0x0006
        rjmp    TIMER1_CMP

INIT:
        ; Initializare stiva
        ldi     R16,low(RAMEND)
        ldi     R17,high(RAMEND)
        out     SPL,R16
        out     SPH,R17

        ; Initializeaza portul A ca iesiri
        ldi     R16,0xFF
        out     DDRA,R16

        ; Limita de numarare
        ldi     R16,low(VAL0-1)
        ldi     R17,high(VAL0-1)
        out     OCR1AH,R17
        out     OCR1AL,R16

        ; Configurare TIMER1 in mod numarare pina la valoarea OCR1A
        ldi     R16,0x00
        out     TCCR1A,R16
        ldi     R16,0x09
        out     TCCR1B,R16

        ; Activeaza intreruperea de comparare pe canalul A
        ldi     R16,0x10
        out     TIMSK,R16
        ; Activeaza sistemul de intreruperi
        sei
        ; Bucla principala a programului
START:
        rjmp    START                   ;ruleaza in bucla pana cand apare o intrerupere

        ; Procedura de tratare a intreruperii de comparatie pe canalul A
TIMER1_CMP:
        ; Salvare context program
        push    R17
        push    R16
        in      R16,SREG
        push    R16
        ; Complementeaza starea bitului 0 din portul A
        in      R16,PORTA
        ldi     R17,0x01
        eor     R16,R17
        out     PORTA,R16

        sbrs R16,0
        rjmp ZERO
        rjmp UNU

ZERO:
        ldi     R16,low(VAL0-1)
        ldi     R17,high(VAL0-1)
        out     OCR1AH,R17
        out     OCR1AL,R16
        rjmp END

UNU:
        ldi     R16,low(VAL1-1)
        ldi     R17,high(VAL1-1)
        out     OCR1AH,R17
        out     OCR1AL,R16

END:
        ; Refacere context program
        pop     R16
        out     SREG,R16
        pop     R16
        pop     R17
        reti
```