

# MIKROPROCESOROVÁ TECHNIKA

Třída: <b>A3</b>	Úloha č. 4	Název <b>Součet více jednobytových čísel</b>	
Jméno: <b>Jakub Tenk</b>	Datum zadání: <b>5. 11. 2020</b>	Datum odevzdání: <b>17. 11. 2020</b>	Známka:

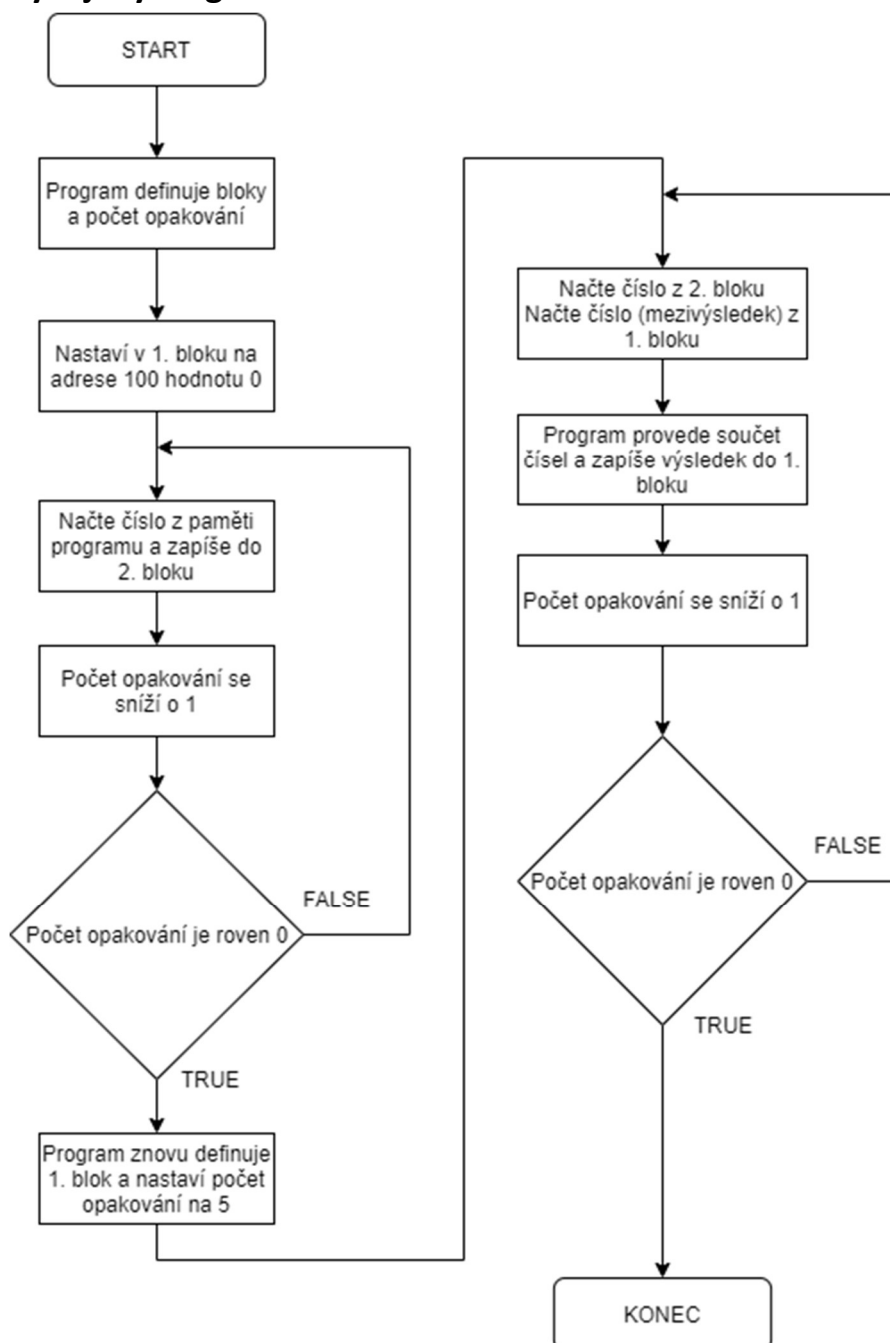
## Zadání:

Vytvořte program v JSA pro sčítání více (minimálně 5) jednobytových čísel. Čísla budou uvedena v tabulce (využítí pseudoinstrukce DB). Uložení výsledku bude od adresy 100.

## Postup:

1. Program definuje bloky, nastaví počet opakování na 5 a nastaví v paměti na adrese 100 hodnotu 0.
2. V cyklu skok1 načte číslo a запиše jej do prvního bloku a sníží počet opakování o 1 (tento cyklus opakuje pětkrát).
3. Počet opakování se nastaví na 5 a přejde do cyklu soucet.
6. V cyklu soucet načte číslo z druhého bloku a načte mezivýsledek z 1. bloku. Provede součet a запиše jako výsledek do 1. bloku. Počet opakování sníží o 1 (tento cyklus opakuje pětkrát).
7. Program se ukončí.

## Vývojový diagram:



## Ukázka z kódu:

```
; >>                                     <<
; >> Součet více jednobytových čísel <<
; >>   Vytvořil Jakub Tenk             <<
; >>                                     <<

.nolist ;neprovádí výpis na obrazovku (debug)
.include "m128def.inc"
.list ;po include začne znovu výpis na obrazovku (debug)

.def poc_opak = R22
.def clear_bl = R23

.cseg

    ldi XL, 0x00    ; >> Definování 1. bloku pro ukládání mezivýsledků a konečného výsledku
    ldi XH, 0x01    ; >>
    ldi YL, 0x00    ; >> Definování 2. bloku pro ukládání načtených čísel z programové paměti
    ldi YH, 0x02    ; >>
    ldi poc_opak, 5  ; >> Nastavení hodnoty počtu opakování (3x)
    ldi clear_bl, 0x00 ; >> Nastavení hodnoty, která se nastaví na začátku v paměti na adrese 100
    st X,clear_bl   ; >> Nastavení hodnoty 0 v paměti na adrese 100

    ldi ZL, LOW(cislo<<1) ; >> odkázání ukazatele Z do programové paměti
    ldi ZH, HIGH(cislo<<1) ; >> na místo (adresa) kde se nachází cislo1

skok1:
    lpm R16, Z+
    st Y+,R16      ; >> Zápis 1. čísla do 1. bloku
    dec poc_opak
    brne skok1

    ; >> Resetování hodnoty počtu opakování, nastavení na 3
    clr poc_opak
    ldi poc_opak, 3
    ; >>

    ldi YL, 0x00    ; >> Znovu definování 2. bloku pro ukládání načítání čísel
    ldi YH, 0x02    ; >>

soucet:
    ld R16, Y+      ; >> načtení čísla
    ld R17, X        ; >> načtení mezivýsledku z paměti na adrese 100 (při začátku programu je hodnota 0)

    add R17, R16     ; >> R17 = R17+R16
    st X,R17         ; >> Uložení součtu do paměti na adresu 100

    dec poc_opak
    brne soucet

cislo: .db 0x01, 0x02, 0x03, 0x04, 0x05
konec: rjmp konec
```

## Závěr:

Tento program splňuje funkčnost podle zadání, sečte všechna zadaná jednobytová čísla a uloží výsledek. Tato úloha byla podle mě mnohem lehčí než předešlá, protože jsem mohl vycházet z kostry starého programu a jen ho upravit. Což jsem také udělal, takže díky splnění úloze číslo tři, mi to tento program trval sestavit deset minut.