

Hardverközeli programozás 1

9. gyakorlat

Kocsis Gergely
2019.04.08.

Két szám összeadása

Írj olyan DIY Calculator Assembly programot, mely a billentyűzetről beolvas két 256-nál kisebb pozitív egész számot 10-es számrendszerben, majd kiszámolja a két szám összegét.

A program jelenítse meg mind a beolvasott számokat, mind az eredményt 10-es számrendszerben. A két szám közé írjon egy “+” jelet (\$2B) az eredmény elé pedig “=”-t (\$3D).

```
#####  
## Start of main program body                                ##  
#####  
      JSR      [READNUM]          # Elso szam beolvasasa  
      JSR      [DISPNUM]         # Elso szam kiirasa  
  
      LDA      $2B  
      STA      [MAINDISP]  
  
      JSR      [READNUM]         # Masodik szam beolvasasa  
      JSR      [DISPNUM]         # masodik szam kiirasa  
  
      LDA      $3D  
      STA      [MAINDISP]  
  
      JSR      [ADDNUM]          # Eredmeny kiszamitasa  
      JSR      [DISPNUM]         # Eredmeny kiirasa  
  
      JMP      [$0000]  
  
#####  
## End of main program body                                ##  
#####
```

Két szám összeadása

```
READNUM: PUSHSR
          STA      [R_ACC] # ACC elmentése
          POPA
          STA      [R_SR]  # SR elmentése
          POPA
          STA      [R_RADDR]      # RADDR MS elmentése
          POPA
          STA      [R_RADDR+1]    # RADDR LS elmentése
# -----
```

READNUM szubrutin törzse

```
# -----
          LDA      [R_RADDR+1]    # RADDR LS a verembe
          PUSHA
          LDA      [R_RADDR]      # RADDR MS a verembe
          PUSHA
          LDA      [R_SR]
          PUSHA                    # SR a verembe
          LDA      [R_ACC]        # ACC visszaállítása
          POPSR                    # SR visszaállítása
          RTS
# -----
          R_ACC:   .BYTE
          R_SR:    .BYTE
          R_RADDR: .2BYTE
```

Két szám összeadása

Számjegyek beolvasása

```
R_LOOP:      BLDX    $FFFF
              LDA     [KEYPAD]
              JN      [R_LOOP]
              CMPA   $13             # Enterig számokat olvasok
              JZ     [R_KESZ]
              CMPA   $09
              JC     [R_LOOP]
              INCX
              STA    [SZAM,X]
              JMP    [R_LOOP]
```

Két szám összeadása

Átváltás hexadecimális formára:

```
R_KESZ: INCX
        LDA    $FF
        STA    [SZAM,X]      # SZAM VEGE KARAKTER A SZAM TOMBBE
        BLDX   0
        LDA    0
        STA    [R_RES]      # 0-rol indul a szam
R_LOOP2: LDA    [SZAM,X]
        JN     [R_VEGE]     # Ha vegjelet olvasok vege
        STA    [R_TEMP]     # Az uj szamjegy elmentese
        LDA    [R_RES]
        SHL
        STA    [R_MUL2]     # szorzas 2-vel
        SHL
        SHL                 # szorzas 8-cal
        ADD    [R_MUL2]     # Az ACC-ben van a 10-zel szorzott eddigi ért.
        ADD    [R_TEMP]     # Az uj szamjegy hozzaadasa
        STA    [R_RES]
        INCX
        JMP    [R_LOOP2]

R_VEGE: LDA    [R_RES]
        PUSHA
```

Két szám összeadása

Megjelenítés (százask):

```
POPA
PUSHA # Kiolvasom a megjeleníteni valót
CMPA 199 # ha nagyobb, mint 199...
JNC [C100]
PUSHA
LDA 2 # ...kiírok egy kettést
STA [MAINDISP]
STA [D_FLAG]
POPA
SUB 200
```

```
C100: CMPA 99 # ha nagyobb, mint 99...
JNC [C90]
PUSHA
LDA 1 # ...kiírok egy egyest
STA [MAINDISP]
STA [D_FLAG]
POPA
SUB 100
```

Két szám összeadása

Megjelenítés (tizesek, egyesek):

C90:	BLDX	0	# Osszeszamolja hany tizes kell
DN_LOOP:	CMPA	9	
	JNC	[C90_D]	
	INCX		
	SUB	10	
	JMP	[DN_LOOP]	
C90_D:	STA	[D_TEMP8]	
	BSTX	[D_TEMP16]	
	LDA	[D_TEMP16+1]	
	JZ	[D_NULLA]	
	STA	[MAINDISP]	# Tizesek kiirasa ha nem 0
	JMP	[D_EGYES]	
D_NULLA:	LDA	[D_FLAG]	# Tizesek kiirasa, ha 0
	JZ	[D_EGYES]	
	LDA	0	
	STA	[MAINDISP]	
D_EGYES:	LDA	[D_TEMP8]	
	STA	[MAINDISP]	# Egyesek kiirasa
	JMP	[D_VEGE]	

Két szám összeadása

Összeadó szubrutin:

```
ADDNUM:          PUSHSR
                 STA     [A_ACC]          # ACC elmentése
                 POPA
                 STA     [A_SR]           # SR elmentése
                 POPA
                 STA     [A_RADDR]        # RADDR MS elmentése
                 POPA
                 STA     [A_RADDR+1]      # RADDR LS elmentése
# -----
                 POPA
                 STA     [A_TEMP8]
                 POPA
                 ADD     [A_TEMP8]
                 PUSHA
# -----
A_VEGE:          LDA     [A_RADDR+1]      # RADDR LS a verembe
                 PUSHA
                 LDA     [A_RADDR]        # RADDR MS a verembe
                 PUSHA
                 LDA     [A_SR]
                 PUSHA                    # SR a verembe
                 LDA     [A_ACC]          # ACC visszaállítása
                 POPSR                    # SR visszaállítása
                 RTS
```


Két szám összeadása

<http://irh.inf.unideb.hu/~kocsisg/10num.asm>