

# Példák vezérlés átadásra

## Feltételes vezérlés átadás

Írjunk programrészletet a  $sign(x)$  függvény megvalósítására!

$$sign(x) = \begin{cases} -1, & \text{ha } x < 0 \\ 0, & \text{ha } x = 0 \\ 1, & \text{ha } x > 0 \end{cases}$$

Feltételezhetjük, hogy az operandus az adatszegmensben egy  $x$  címkével van ellátva és 16 bites előjeles érték.

---

```
MOV AX, x      ; az x értéke legyen AX-ben
CMP AX, 0      ; összehasonlítjuk AX-et 0-val
MOV AX, 0      ; a MOV utasítás nincs
               ; hatással a flag-ekre
JE vege       ; ha 0, akkor ugras
               ; a vege címkere
JG poz        ; ha pozitív, akkor ugras
               ; a poz címkere
JL neg        ; ha negatív, akkor ugras
               ; a neg címkere

poz:  AX = 1d
      JMP vege      ; ugrani kell a vegere
neg:  AX = -1d
vege:  ...
```

---

## Ciklus szervező utasítások

Számoljuk ki az első 10 pozitív egész szám összegét! Az eredményt AX-ben kapjuk.

---

```

MOV CX,10d    ; szamlalo beallitasa
MOV AX,0      ; kezdo ertek
ciklus: ADD AX, CX    ; kiszamoljuk a reszosszeget
        LOOP ciklus  ; ha CX ≠ 0, akkor folytassuk
vege:    ...

```

---

## Fealdatok

1. Számoljuk ki a következő kifejezések értékét, az eredményt az *AX*-ben kapjuk. Az adatszégmensben a következő adatok találhatóak:

---

```

adat segment
a db 4 ; elojeltelen
b db 3 ; elojeltelen
c dw 9 ; elojeltelen
adat ends

```

---

- (a)  $AX := \min(a+b, a+c)$
  - (b)  $AX := \text{mid}(a, b, c)$
  - (c)  $AX := (a+b) == c ? a : b+c$
  - (d)  $AX := \text{abs}(a+b)$
2. Írjunk kódrészletet a következő függvények megvalósítására! Feltehető, hogy a paraméterek az adatszégmensben vannak eltárolva.
    - (a)  $fakt(n)$  ( $n!$ )
    - (b)  $fib(n)$  (Fibonacci sorozat, feltételezhetjük, hogy egy kiíró eljárás kiírja az *AX*-ben lévő előjel nélküli számot.)