

Sztringek és tömbök kezelése

Stringek kezelése

A képernyőre történő kiírás valójában mindig karakteresen történik. A sztringek számok kiírása során először az adatot karakterek sorozatává konvertáljuk, majd az adatokat egymás utsán sorban kiírjuk a képernyőre.

Egyetlen karakter kiírását az Irvine függvénykönyvtár használatával a `WriteChar` függvény segítségével oldható meg. Az eljárás a paraméterét az `AL` regiszterben várja.

```
;Egyetlen betu kiirasa
        CALL AL, 'A'      ; Karakterkód beállítása
        CALL WriteChar   ; Kiírás
```

Egy szöveg kiírásánál feltételezhetjük, hogy vagy a karakterek számát, vagy pedig egy záró karaktert ismerjük. Utóbbira példa az alábbi eljárás. Feltételezzuk, hogy a 0 lesz a záró karakter. A szöveget az adatterületen adjuk meg, és az eljárás meghívása előtt `ESI`-t a szöveg kezdő karakterére kell állítani.

```
;eljaras ami kiir egy 0-val zarodo karakterlancot
;a string elejet esi mutatja
strkiir    proc
kov:       mov al, [esi]    ;kovetkezo betu
           cmp al, 0       ;0 jelzi a str veget
           je strvege      ;ha elertunk a vegere ugrunk
           call WriteChar  ;egy betu kiirasa
           inc esi         ;mutato a kovetkezo beture
           jmp kov         ;ugras

strvege:
           ret             ;eljaras vege

strkiir    endp
```

A honlapon található egyszerű példa program `haho.asm` bemutatja, hogy hogyan is működik ez:

Feladatok

1. Számoljuk meg, hogy hány darab "a" betű van egy karakterláncban!
2. Cseréljünk le egy string minden "a" betűjét "b" betűre.
3. Titkosítás: Cseréljünk ki minden betűt (A-Z, és a-z) az "x" betűre, az egyéb karaktereket hagyjuk meg.
4. Cseréljünk le egy string minden kisbetűjét nagybetűre (UPPER CASE). Mit kell módosítani a programon, hogy a nagybetűkből legyenek kisbetűk? (lower case)
5. Invertáljuk a kis és nagy betűket egy karaktersorozatban. (invert case)
6. A szavak első betűjét állítsuk nagybetűsre, a többi részét pedig kisbetűsre. A szavakat szóközök választják el. (Proper Case)

Tömbök kezelése

A tömbök kezelése a sztringekhez hasonlóan működik. Ilyen esetben viszont a tömb sokszor nem a véget záró értékből tudjuk, hanem közvetlenül megkapjuk a tömbelemek hosszát.

Az alábbi példa egy tömböt, és annak bejárását mutatja be.

```
.data
    T      SDWORD  1, 2, 3, -56, 1956, 847      ; Maga a tömb
    N      DWORD   6                          ; A tömb hossza
...

.Code
...

    MOV     ESI, offset T      ; ESI a tömb kezdetére
    MOV     ECX, N             ; Tömb hosszának betöltése
    MOV     EAX, 0             ; EAX kezdetben nulla
```

```
ciklus:                                ; Ciklus kezdete

MOV     EBX, [ESI]                      ; Aktuális érték beöltése
ADD     ESI, 4                          ; ESI a következő értékre
ADD     EAX, EBX                        ; Összeadás

LOOP   ciklus                          ; Cissza a ciklus kezdetére
```

Feladatok

1. Adjuk össze a Tömb pozitív elemeit!
2. Számoljuk meg, hogy hány negatív eleme van a tömbnek!
3. Írjuk ki a tömb 0 és 10 közötti elemeit!