

Példák sztringek kezelésére

Sztringek beolvasása billentyűzetről

Eljárás sztring beolvasására a billentyűzetről. A beolvasást BIOS hívásokkal valósítjuk meg. A sztring végét az ENTER (új sor) karakter zárja.

```
; eljárás ami beolvas egy stringet az SI által mutatott bufferbe
strolvas  proc
            push si                ;si mentese
olv:
            mov ah, 1h             ;BIOS eljárás sorszama
            int 21h                ;BIOS interrupt
            cmp al, 0dh            ;ENTER kódja
            je olvasvege
            mov [si], al           ;betű tárolása a memoriában
            inc si                 ;a következő betű
                                    ; eggyel magasabb mem.
                                    ; címre kerüljön
            jmp olv                ; vissza az olvasás elejére

olvasvege:
            mov byte ptr [si], 0   ;a string végét 0-val lezárjuk

            mov ax, 0e0ah          ; ujsor karakter kiírása
            int 10h

            pop si
            ret
strolvas  endp
```

Palindróma ellenőrzés

Az alábbi kódrészlet az adatterületen a `szo` címén lévő és '\$' végjelű karaktersorozatról állapítja meg, hogy palindróma-e vagy sem.

```
adat      segment para public 'DATA'
          szo db 'indulagorogaludni','$'
          igen_valasz db 'A szo palindróma',0
          nem_valasz db 'A szo nem palindróma',0
          kvse db 13, 10, 0
adat      ends
```

```
...
; elokeszites
    mov bx, 0
    mov cx, 0
    mov si, offset szo
    mov dl, [si + bx]
    ; cx a szo vegerol indul,
    ; bx a szo elejerol
veget_keres: cmp dl, '$'           ; itt van-e a vegjel?
             je kezd             ; ha igen, akkor kezdjuk
             ; az elemzest
             inc bx              ; lepjunk a kovetkezo karakterre
             mov dl, [si+bx]
             jmp veget_keres

kezd:    xchg bx, cx             ; cx a vegere mutat,
             ; bx az elejen all
vizsgal: dec cx                 ; eloszor csokkenteneni kell
             mov dl, [si+bx]
             xchg bx, cx         ; cx az elejen all,
             ; bx a vegen van (forditott)

             mov al, [si+bx]
             xchg cx, bx         ; visszacserelem bx es
             ; cx poziciojat
             cmp al, dl          ; az aktualis pozicion a ket
             ; betu megegyezik
             jne nem             ; ha nem egyezik meg,
             ; akkor nem palindróma
```

```

        cmp bx, cx            ; ha ugyanazon a pozicion all
                                ; a ket index, akkor jo
        je igen
        inc bx
        cmp bx, cx            ; ha pont egymas mellett
                                ; allt, akkor jo
        je igen
                                ; cx-et a ciklus elejen
                                ; csokkentjuk
        jmp vizsgal
nem:    mov si, offset szo
        call szot_kiir
        mov si, offset        ; uj sort nyit
        kvse
        call szot_kiir
        mov si, offset
        nem_valasz
        call szot_kiir
        mov si, offset
        kvse
        call szot_kiir        ; uj sort nyit
        jmp vege

igen:   mov si, offset szo
        call szot_kiir
        mov si, offset
        kvse
        call szot_kiir
        mov si, offset
        igen_valasz
        call szot_kiir
        mov si, offset
        kvse
        call szot_kiir
        jmp vege
...

```

Feladatok

1. Fordítsunk meg egy stringet! Írjunk eljárást, amely egy string tartalmát megfordítja. Az eljárás SI regiszterben várja a string elejét. Kiindulási forrás: `strford.asm`.
2. Oldjuk meg a feladatot a verem használata nélkül!
3. Használjuk a vermet a megoldáshoz!
4. Használjuk a sztingkezelő utasításokat