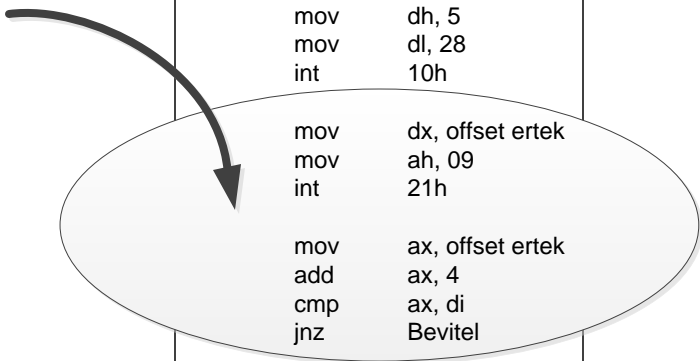
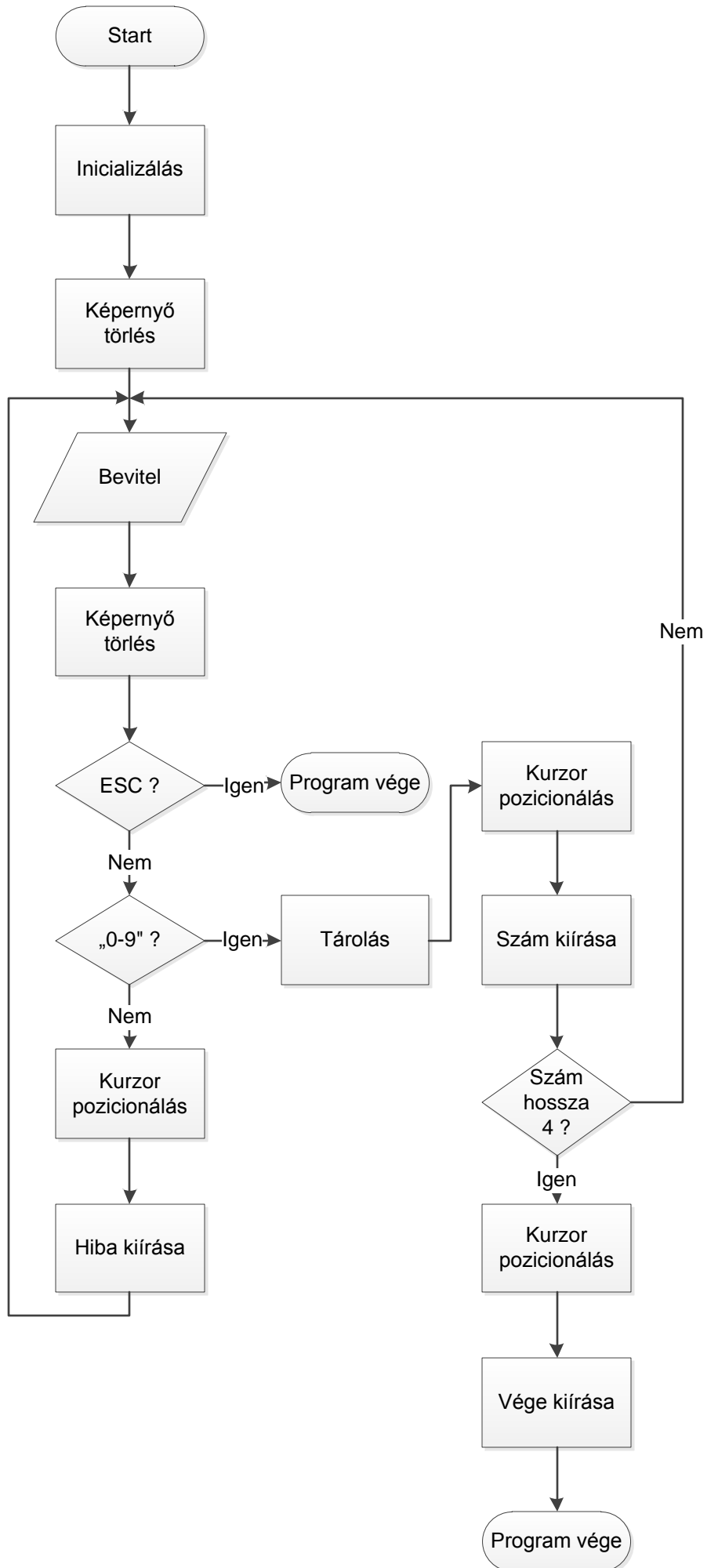


Feladat2: beker.asm

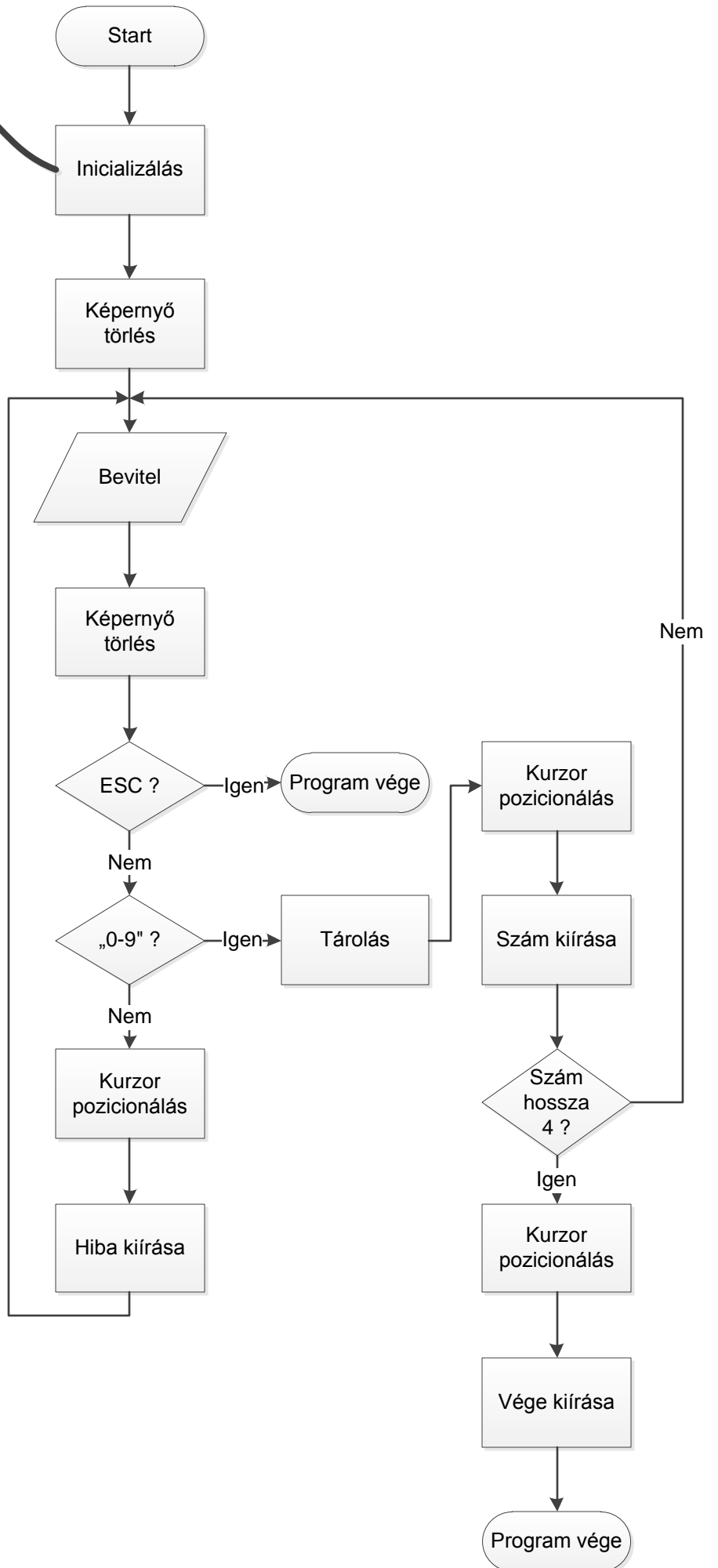
Feladat meghatározása	Implementálás	Implementálás
<p>A program célja, gyakorolni az eddig megismert funkciókat, valamint megismerni a memória tartalmának futásidőben történő módosítását.</p> <p>A program egy négyjegyű decimális számot vár, amit a billentyű segítségével lehet bevinni. A számot egy, a memóriában elhelyezett változó tárolja, karakterenként ASCII kódban.</p> <p>A program törli a képernyőt és csak számokat fogad el. Egyéb gombok megnyomására hibaüzenetet ad.</p> <p>Program bemutatása:</p> <ul style="list-style-type: none"> <li>A programot el kell készíteni a tároló programág kivételével. A tároló ágot egy JMP Bevitel utasítással kell helyettesíteni. Az így lefordított program látszólag semmit sem tesz, ha számot ütünk le, minden egyéb billentyű hatására hibaüzenetet küld. ESC billentyű leütésére a program megáll.</li> <li>A teljesen elkészített program a számok leütése esetén a képernyőre kiírja az eddig begépett számsort. Ha a begépett számok hossza elérte a 4-et, az kiíródik a képernyőre és a program a kilépés ágon folytatódik.</li> </ul>	<p>Code Segment assume CS:Code, DS:Data, SS:Stack</p> <p>Start:</p> <pre> mov ax, Code mov ds, ax  mov di, offset ertek  mov ax, 03 int 10h  Bevitel: xor ax, ax int 16h  mov bx, ax mov ax, 03 int 10h mov ax, bx  cmp al, 27 jz Program_vege  mov cx, 10 mov ah, '0'  Vizsg: cmp al, ah jz Tarol inc ah loop Vizsg  mov ah, 02h mov bh, 0 mov dh, 10 int 10h  mov dx, offset hiba mov ah, 09 int 21h  jmp Bevitel  Tarol: mov [di], al inc di mov al, '\$' mov [di], al  mov ah, 02h mov bh, 0 mov dh, 5 mov dl, 28 int 10h  mov dx, offset ertek mov ah, 09 int 21h  mov ax, offset ertek add ax, 4 cmp ax, di jnz Bevitel  mov ah, 02h mov bh, 0 mov dh, 7 int 10h </pre>	<pre> mov dx, offset uzenet mov ah, 09h int 21h  Program_vege: mov ax, 4c00h int 21h  hiba: db 'Nem megengedett karakter!\$' ertek: db '*****\$' uzenet: db 'Vege a bevitelnek\$'  Code Ends Data Segment Data Ends Stack Segment Stack Ends  End Start </pre>

Optimalizálási lehetőség

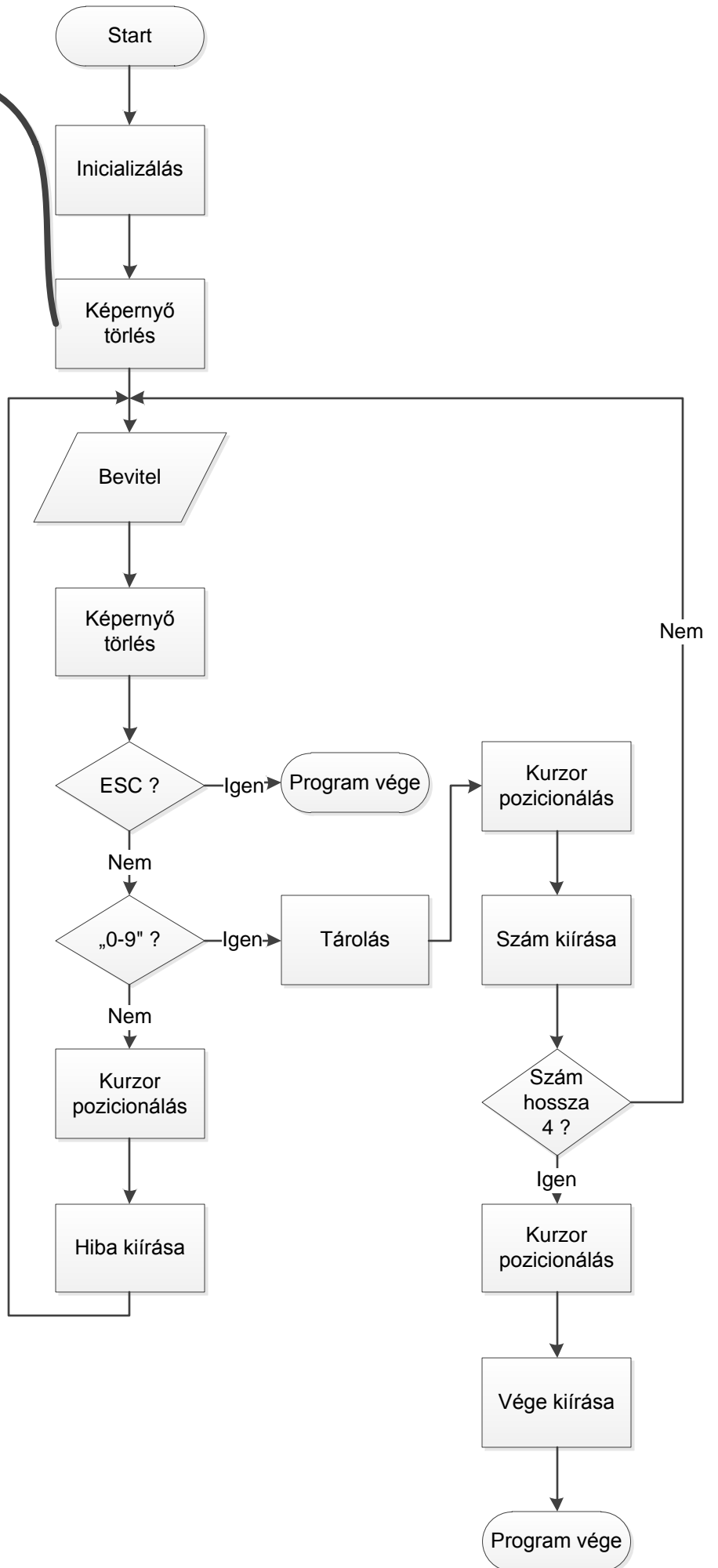




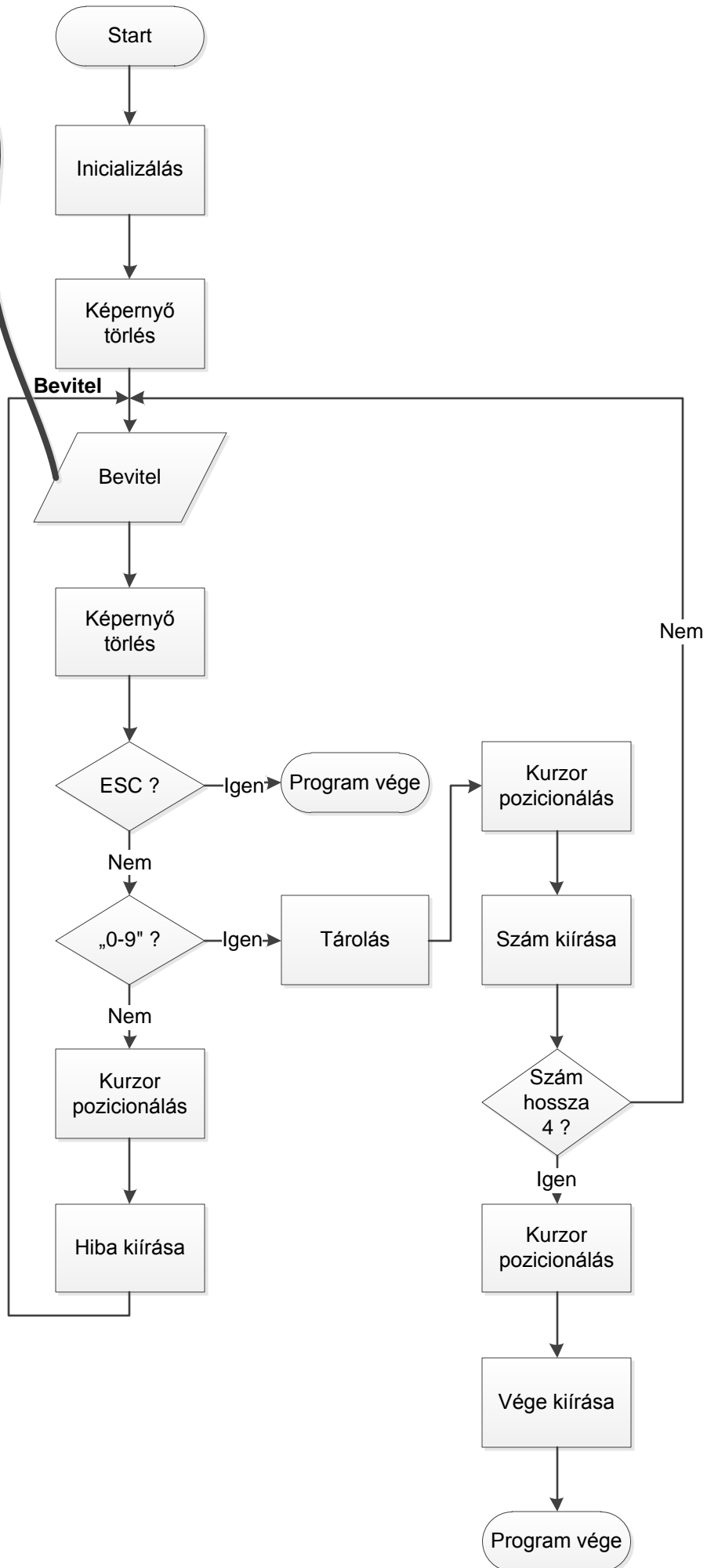
Implementálás	
Inicializálás	
;program vége után, még a Code szegmensbe	
hiba:	db "Nem megengedett karakter!\$"
ertek:	db "****\$"
uzenet:	db "Vége a bevitelnek\$"
-----	
;program elejére	
mov	di, offset ertek



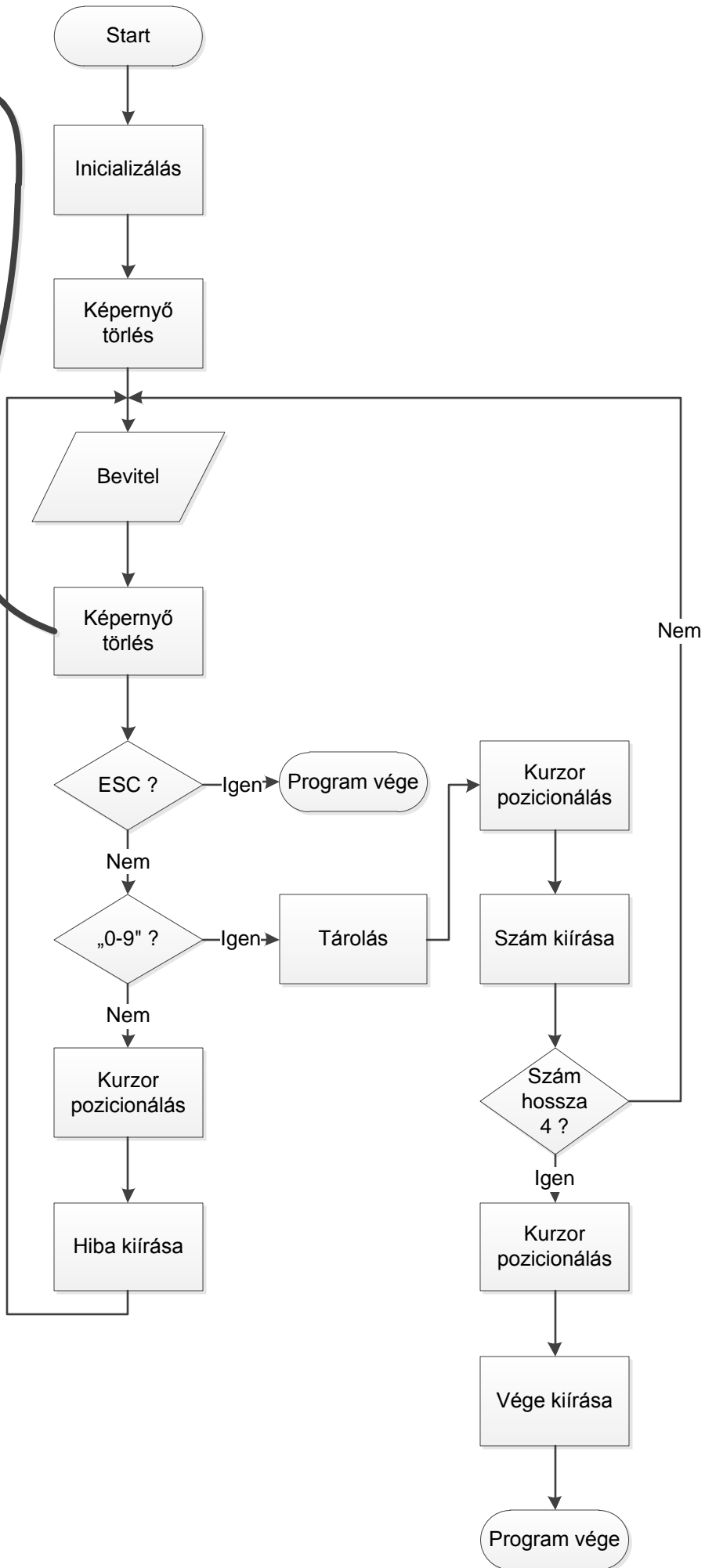
Implementálás	
	Képernyő törlés
mov	ax, 03h
int	10h



Implementálás	
Bevitel	
Bevitel:	
xor	ax, ax
int	16h ;várakozás egy
billentyű leütésére	



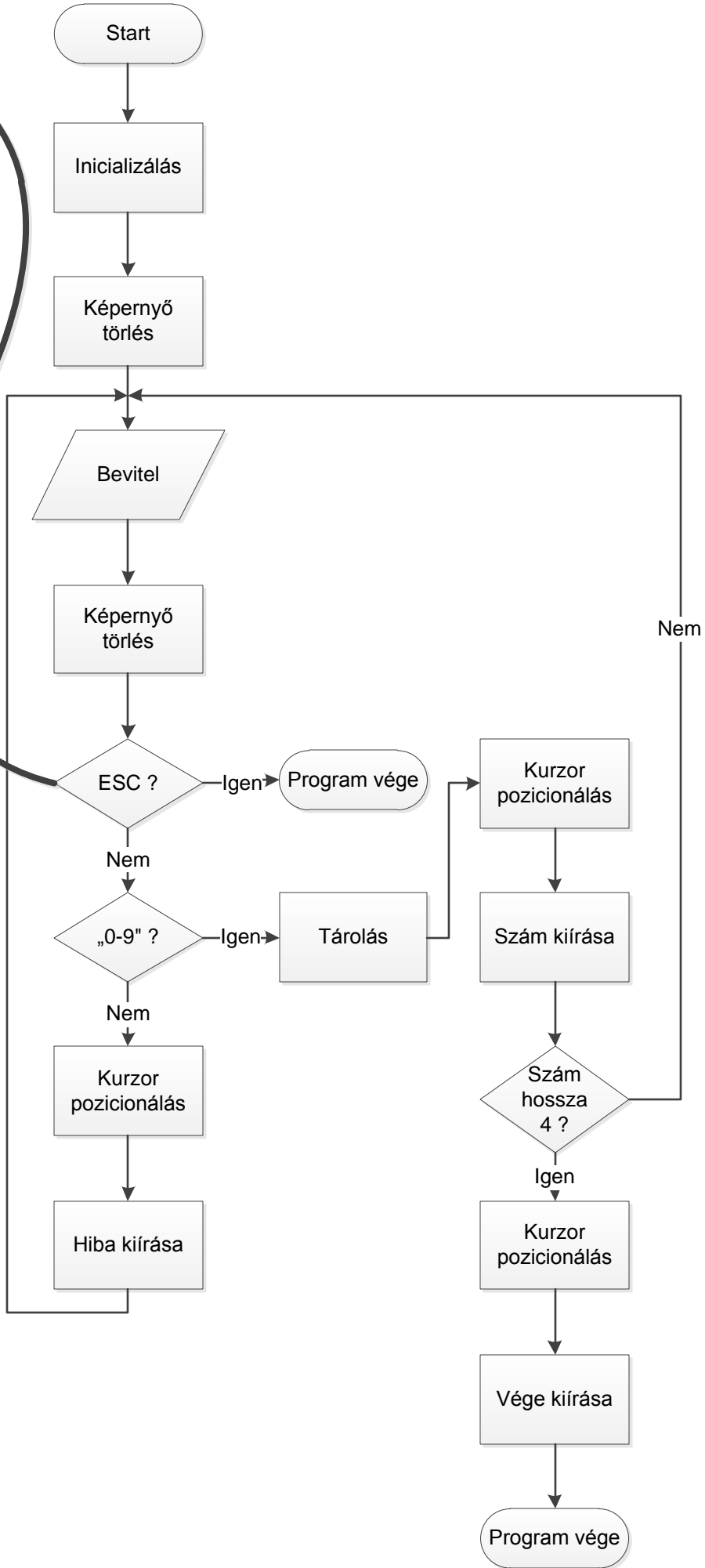
Implementálás		
<b>Képernyő törlés</b>		
mov	bx, ax	;ax elmentése
mov	ax, 03h	
int	10h	
mov	ax, bx	;ax visszaállítása



```

Implementálás
ESC ?
cmp    al, 27
jz     Program_vege ;igaz feltétel
;hamis feltétel esetén fut tovább a
program

```

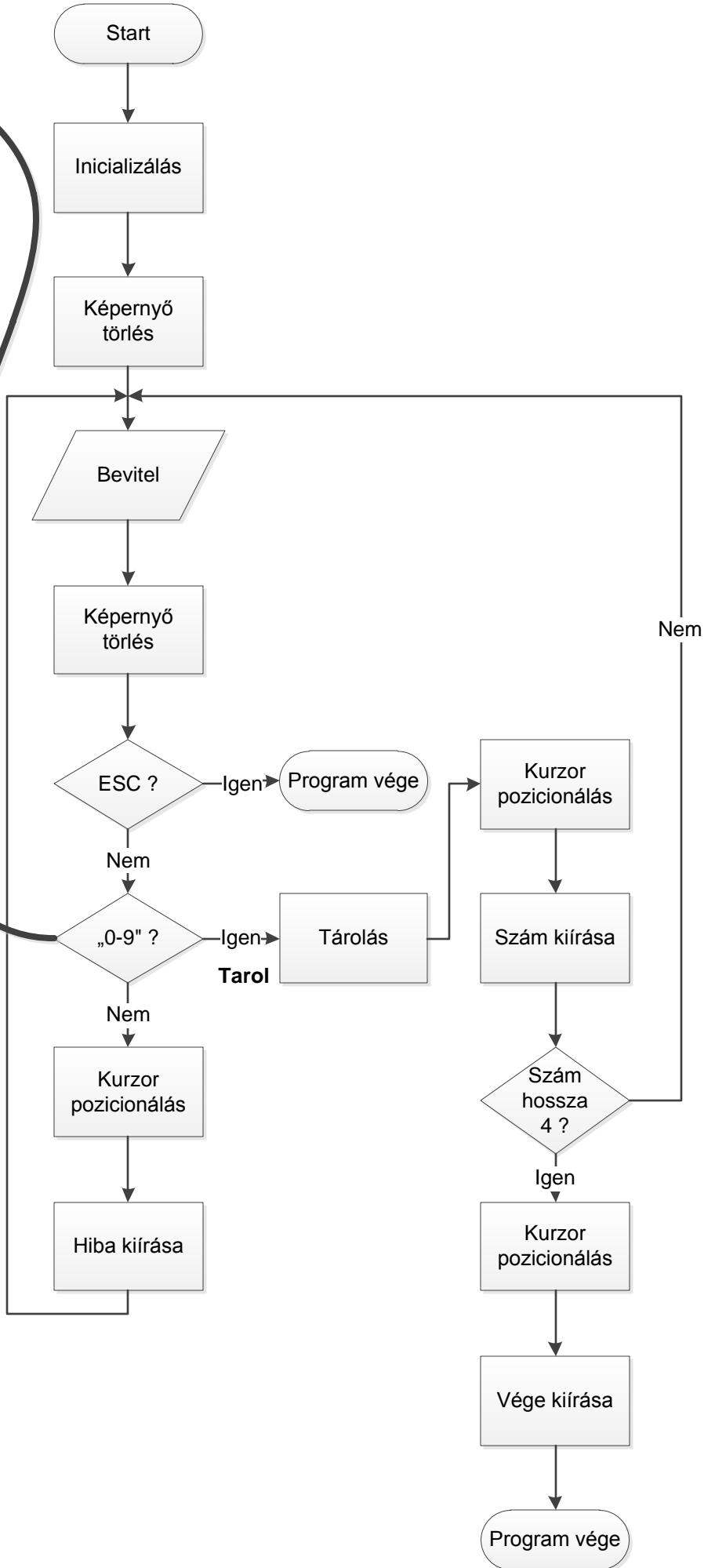


**Implementálás**

```

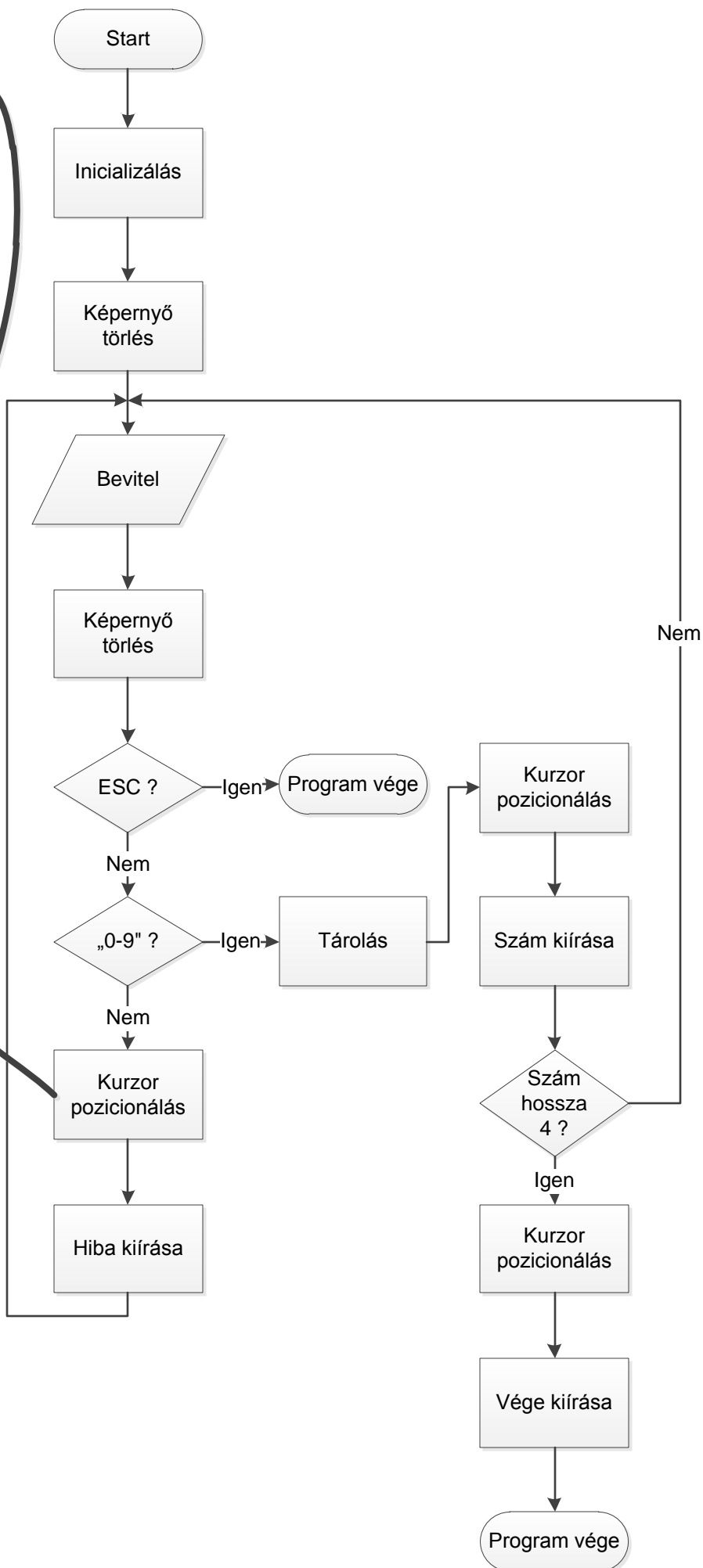
"0-9" ?
mov    cx, 10
mov    ah, "0"
Vizsg:
cmp    al, ah
jz     Tarol
inc    ah
loop  Vizsg
;hamis feltétel esetén fut tovább a program

```

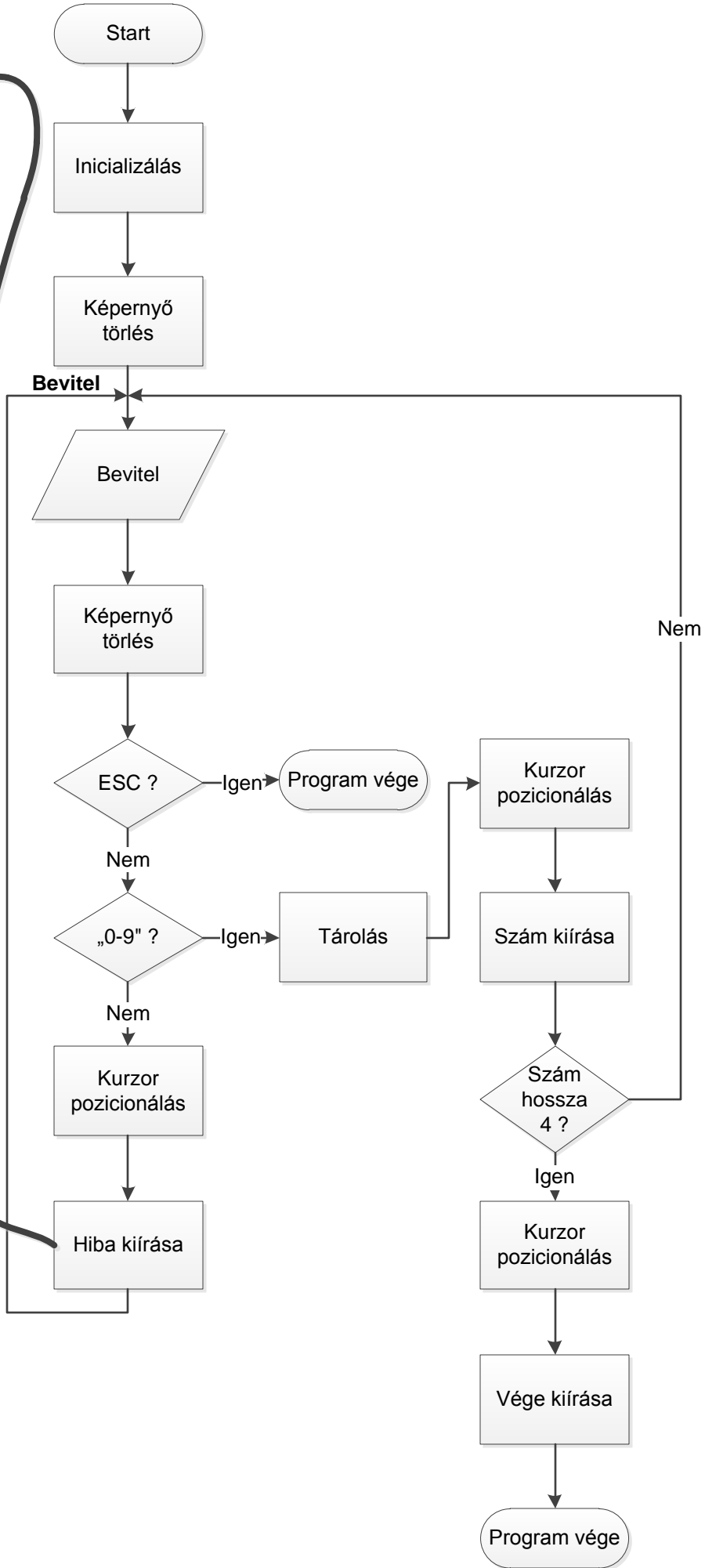




Implementálás	
Kurzor pozicionálás	
mov	ah, 02h
mov	bh, 0
mov	dh, 10
mov	dl, 0
int	10h



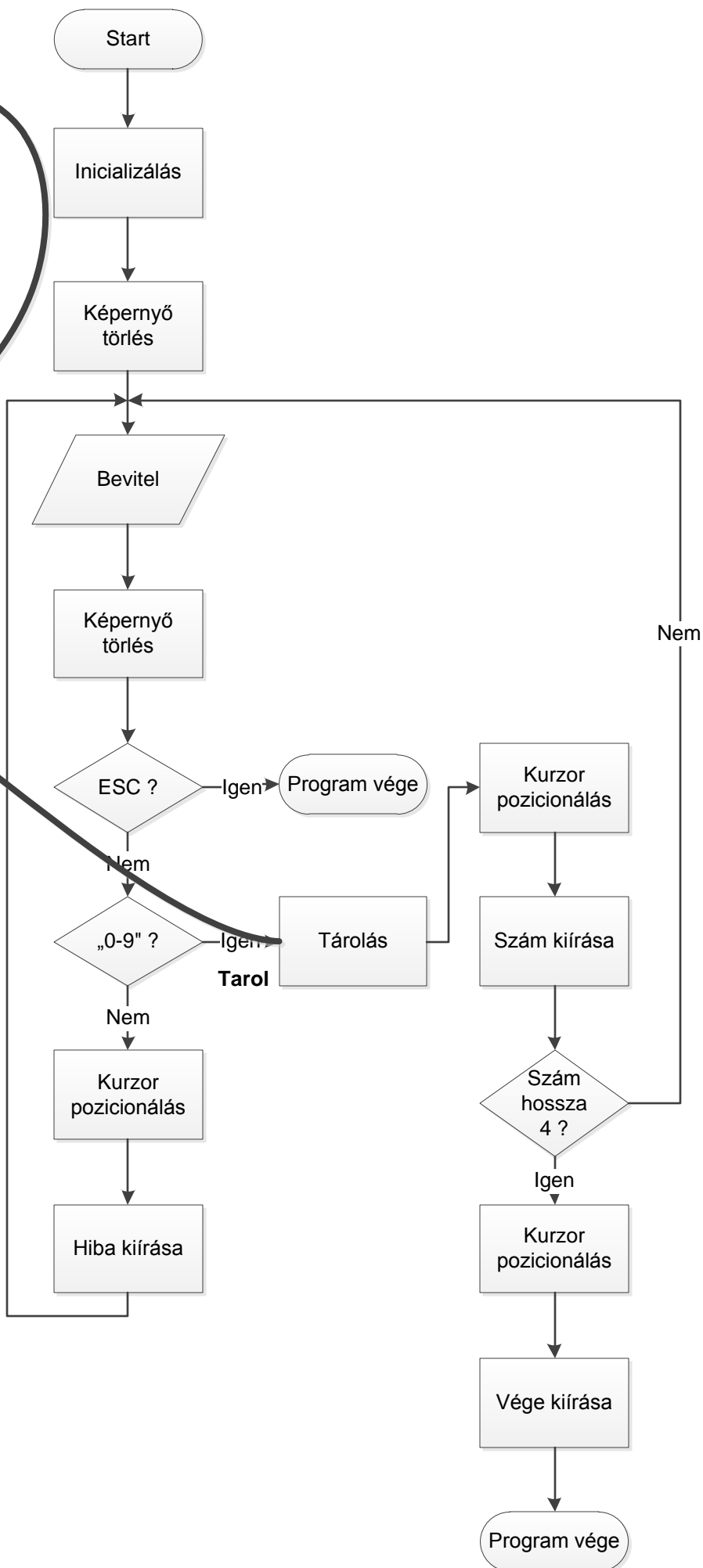
Implementálás	
	Hiba kiírása
mov	dx, offset hiba
mov	ah, 09h
int	21h
jmp	Bevitel



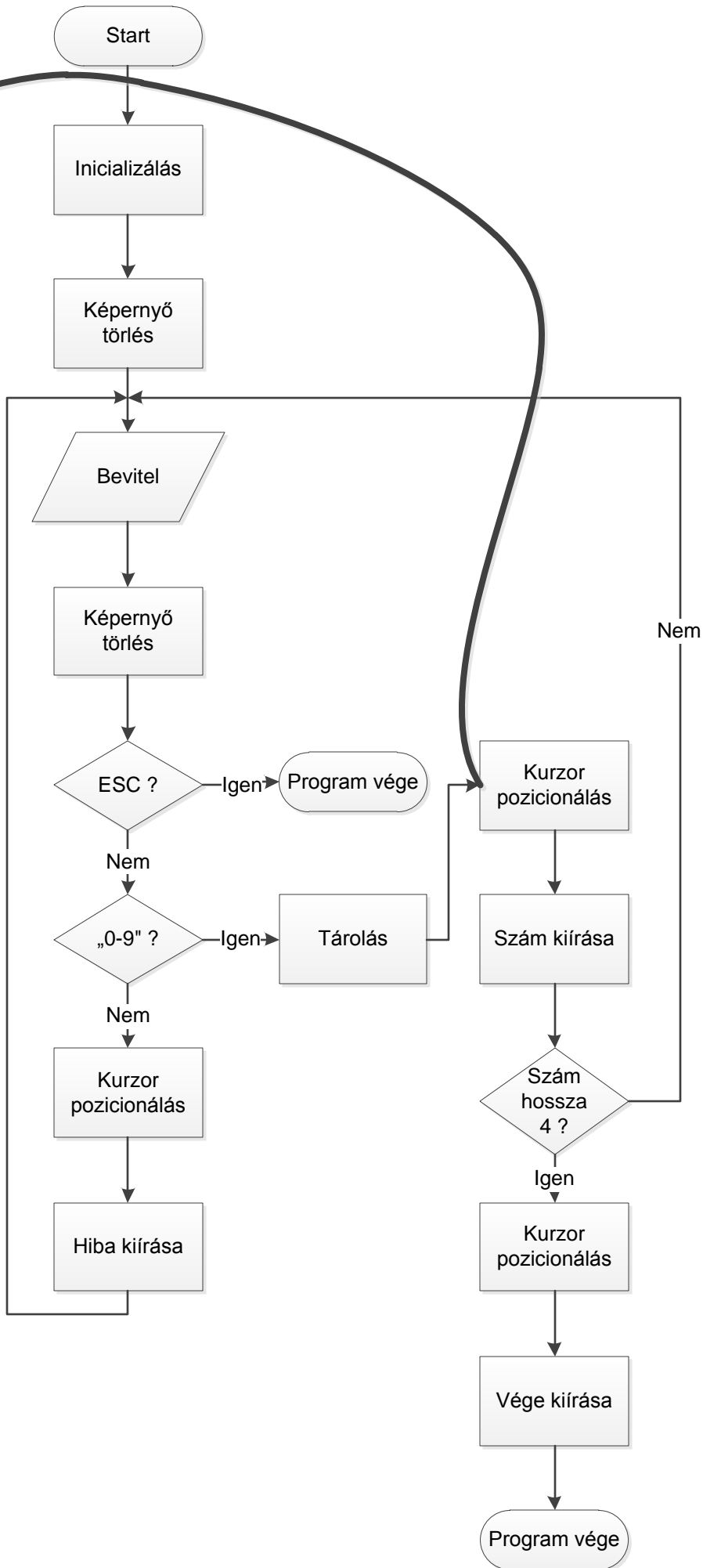
```

Implementálás
Tárolás
Tarol:
;di regiszter tartalma az ertek címke offset
memória címe
mov [di], al ;kiírjuk a leütött billentyű
ASCII értékét a memóriába
inc di ;növeljük az offset címet
mov al, "$"
mov [di], al ;kiírunk egy
karakter sorozatot lezáró vezérlő karaktert

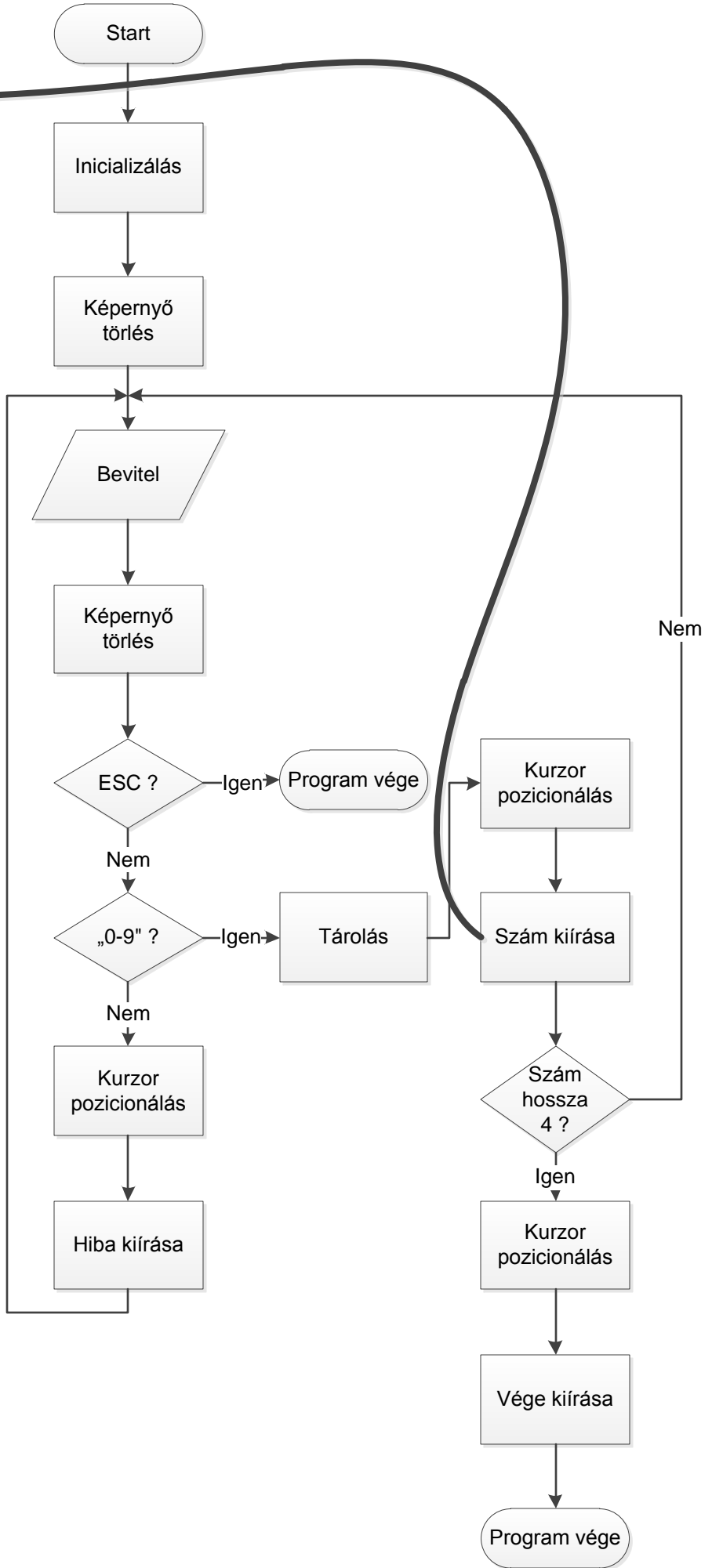
```



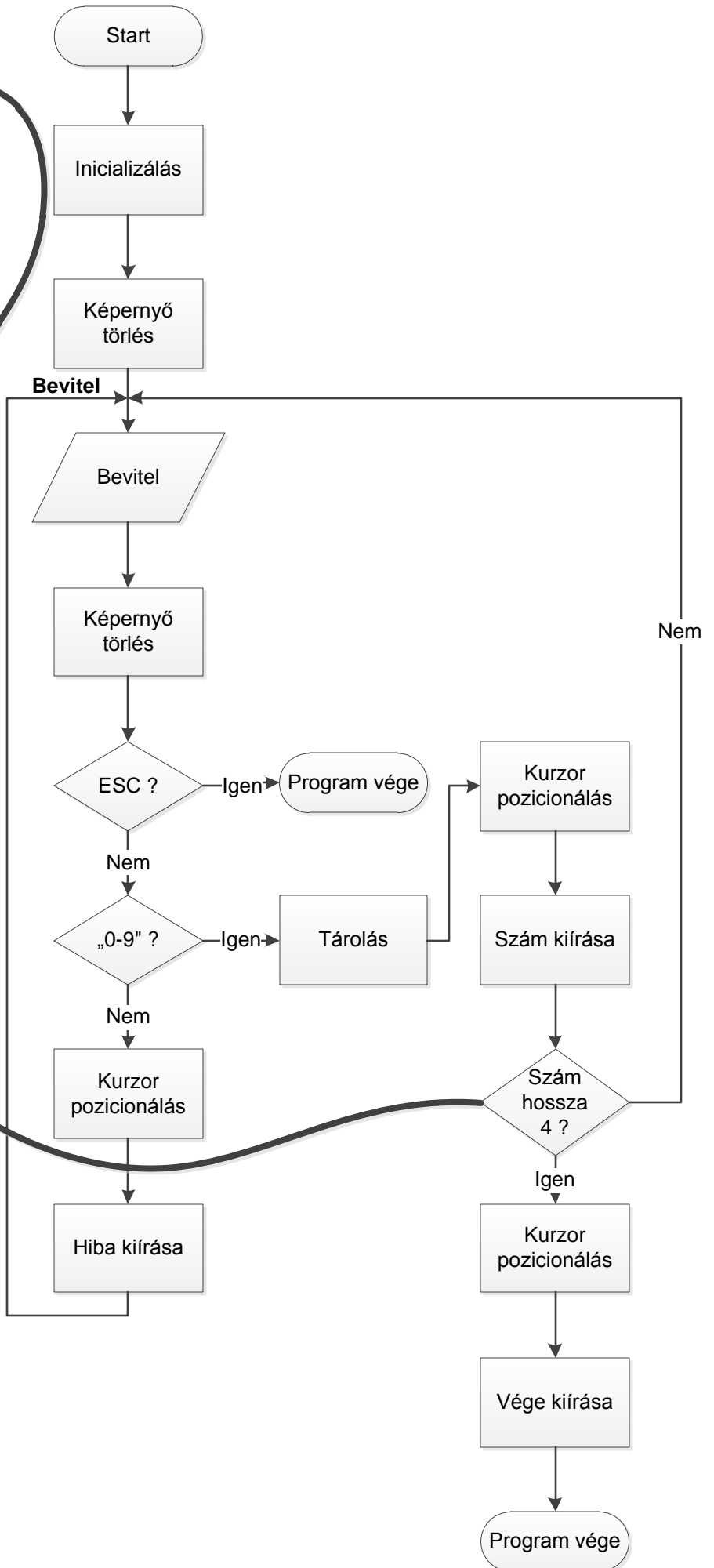
Implementálás	
Kurzor pozicionálás	
mov	ah, 02h
mov	bh, 0
mov	dh, 5
mov	dl, 28
int	10h



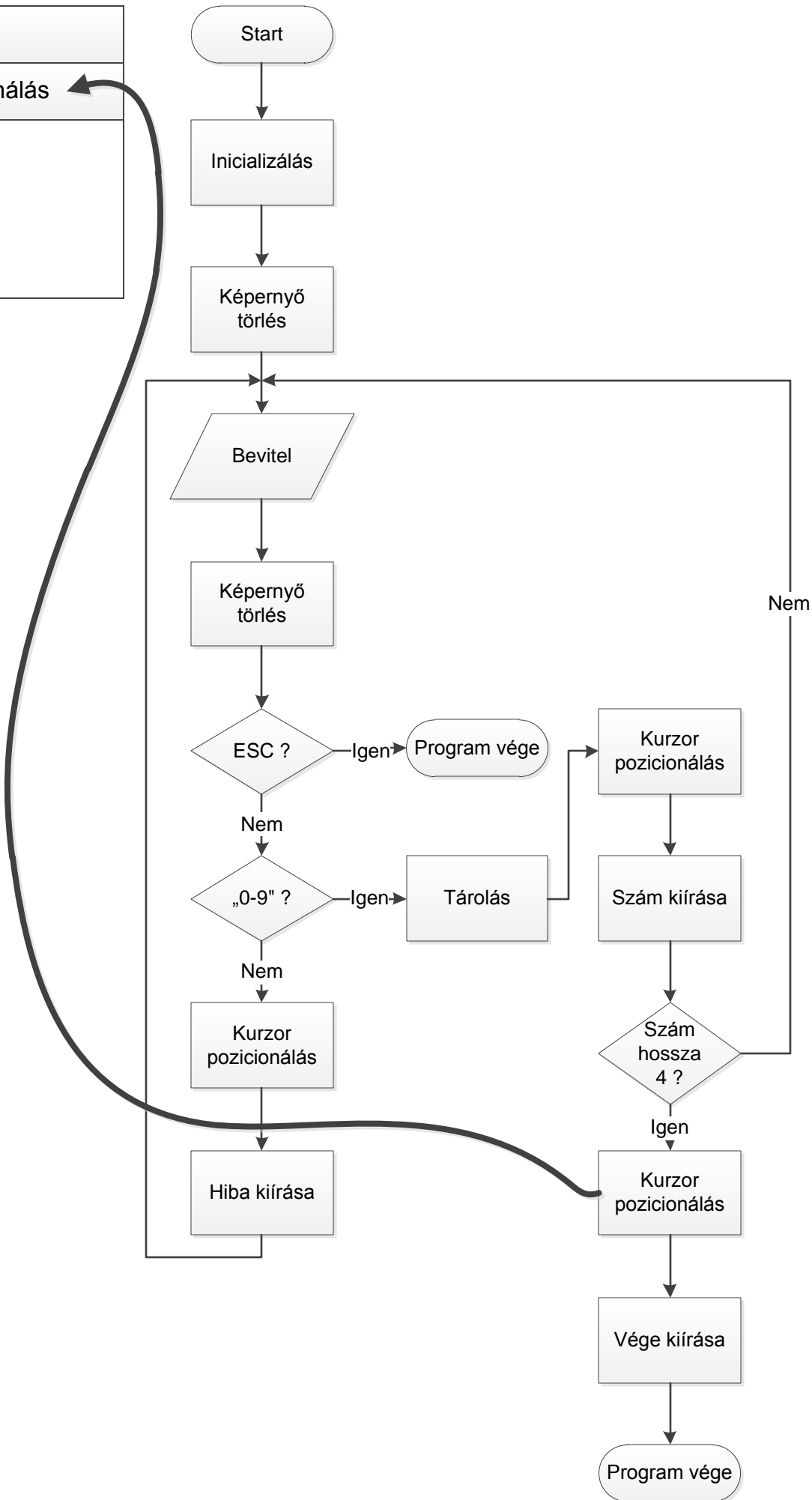
Implementálás	
<b>Szám kiírása</b>	
mov	dx, offset ertek
mov	ah, 09h
int	21h



Implementálás	
	Szám hossza 4 ?
mov	ax, offset ertek
add	ax, 4
cmp	ax, di
jnz	Bevitel ;még nem ütöttünk le 4 numerikus billentyűt, kezdőcím+4 nagyobb mint az utolsó tárolt karakter címe
;egyébként tovább fut a program	



Implementálás	
Kurzor pozicionálás	
mov	ah, 02h
mov	bh, 0
mov	dh, 7
mov	dl, 0
int	10h



Nem

Implementálás	
	Vége kiírása
mov	dx, offset uzenet
mov	ah, 09h
int	21h

