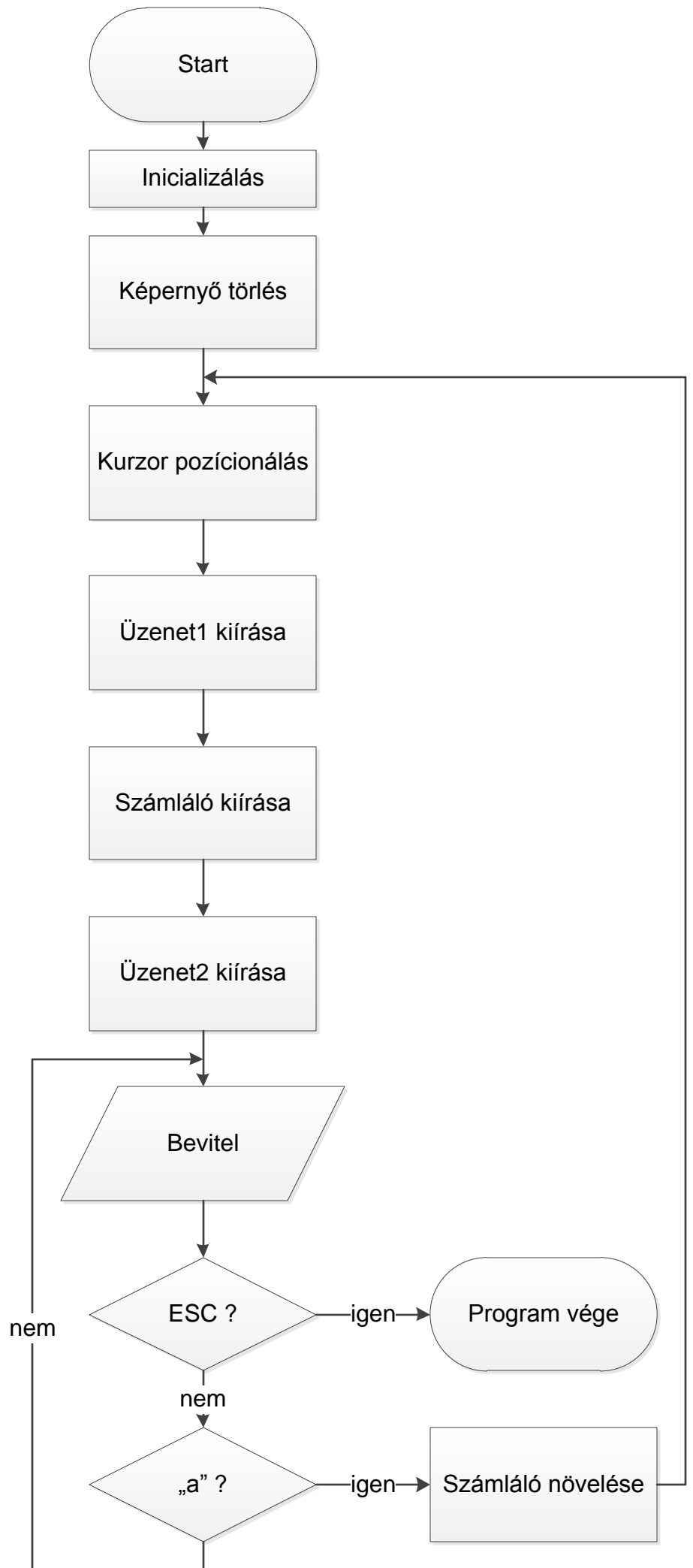


Feladat1: szamlalo.asm

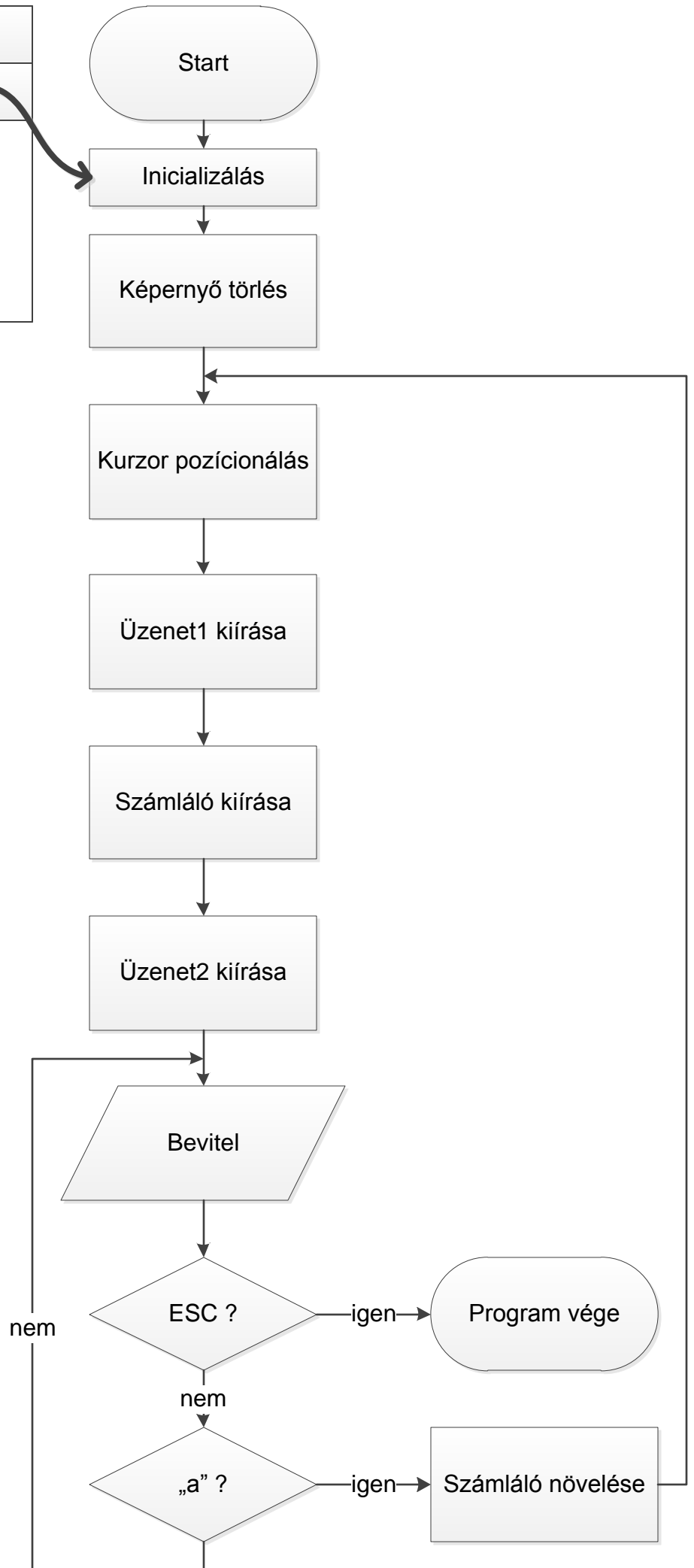
Feladat meghatározása	Implementálás
<p>A program célja, megismertetni az assembly program felépítését. A program kiír a képernyőre egy üzenetet, valamint számlálja az „a” billentyű leütését. A számlálás 0-tól indul, 9 után az ASCII tábla megfelelő jelei kerülnek a képernyőre.</p> <p>Program bemutatása:</p> <ul style="list-style-type: none"> <li>• Egy üres, de fordítható és futtatható keret elkészítése. Ez a keret lesz az alapja a további programoknak is.</li> <li>• A kész keretbe egy üzenet kerül. Ez a klasszikus „HELLO WORLD” program. Ennek segítségével bemutatásra kerül az egyszerű string kiírás az AH09h, Int21h DOS funkcióval.</li> <li>• Létre kell hozni egy programhurkot. Ez egy egyszerű JMP utasítással történik. Biztosítani kell a hurokból való kilépést, hogy ne legyen végtelen ciklus. Ehhez ismertetni kell az AH00h, Int16h funkciót, valamint meg kell írni egy feltételvizsgálatot.</li> </ul>	<pre> Code      Segment assume CS:Code, DS:Data, SS:Stack  Start:     mov     ax, Code     mov     DS, ax      mov     ax, 03h     int     10h  Kiir:     mov     ah, 02h     mov     bh, 0     mov     dh, 10     mov     dl, 0     int     10h      mov     dx, offset uzenet1     mov     ah, 09h     int     21h      mov     dx, offset szamlalo ;mov     ah, 09h     int     21h      mov     dx, offset uzenet2 ;mov     ah, 09h     int     21h  Bevitel:     xor     ax, ax     int     16h      cmp     al, 27     jz      Program_Vege      cmp     al, "a"     jz      Szamol     jmp     Bevitel  Szamol:     mov     di, offset szamlalo     mov     al, [di]     inc     al     mov     [di], al      jmp     Kiir  Program_Vege:     mov     ax, 4c00h     int     21h  uzenet1:     db      "Az a billentyű \$"  uzenet2:     db      " alkalommal volt leutve\$"  szamlalo:     db      "0\$"  Code      Ends  Data      Segment Data      Ends  Stack     Segment Stack     Ends  End       Start </pre>



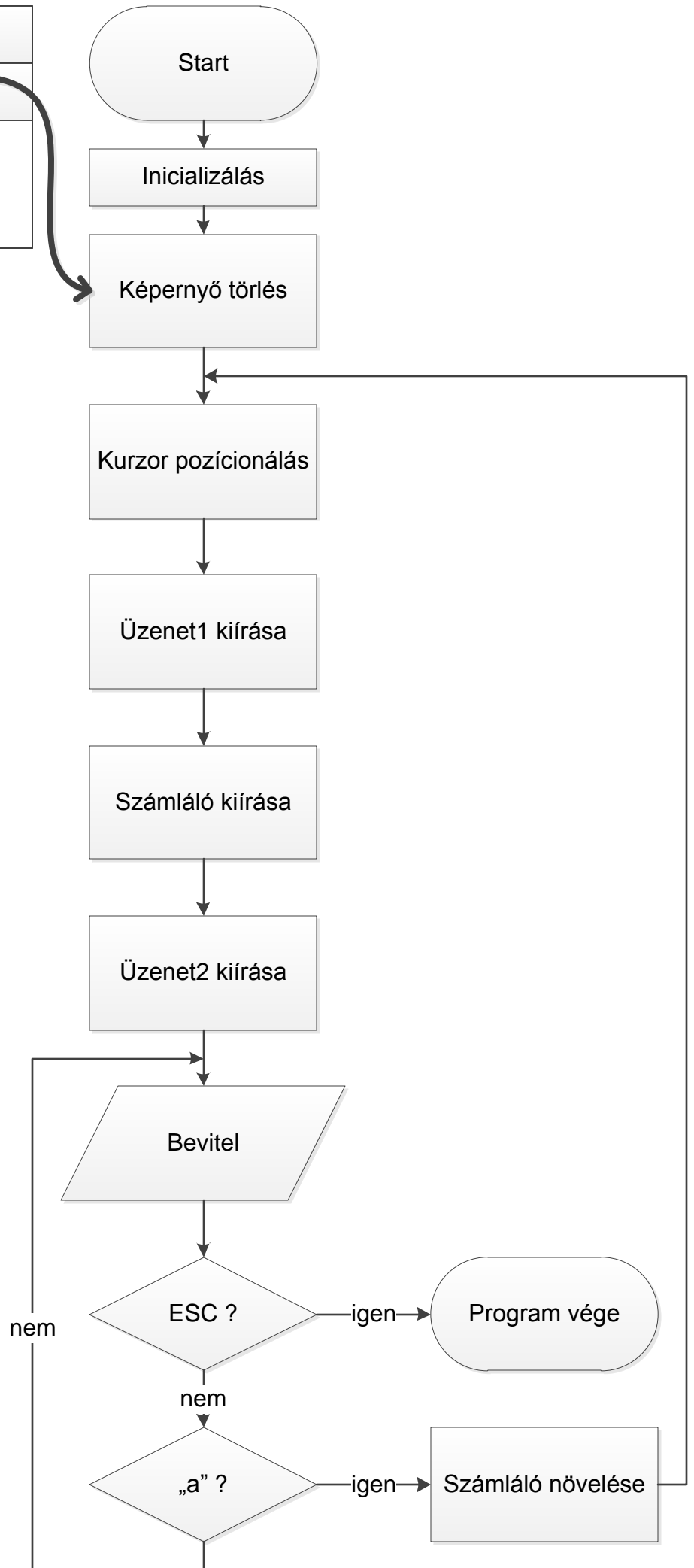
**Implementálás**

Inicializálás

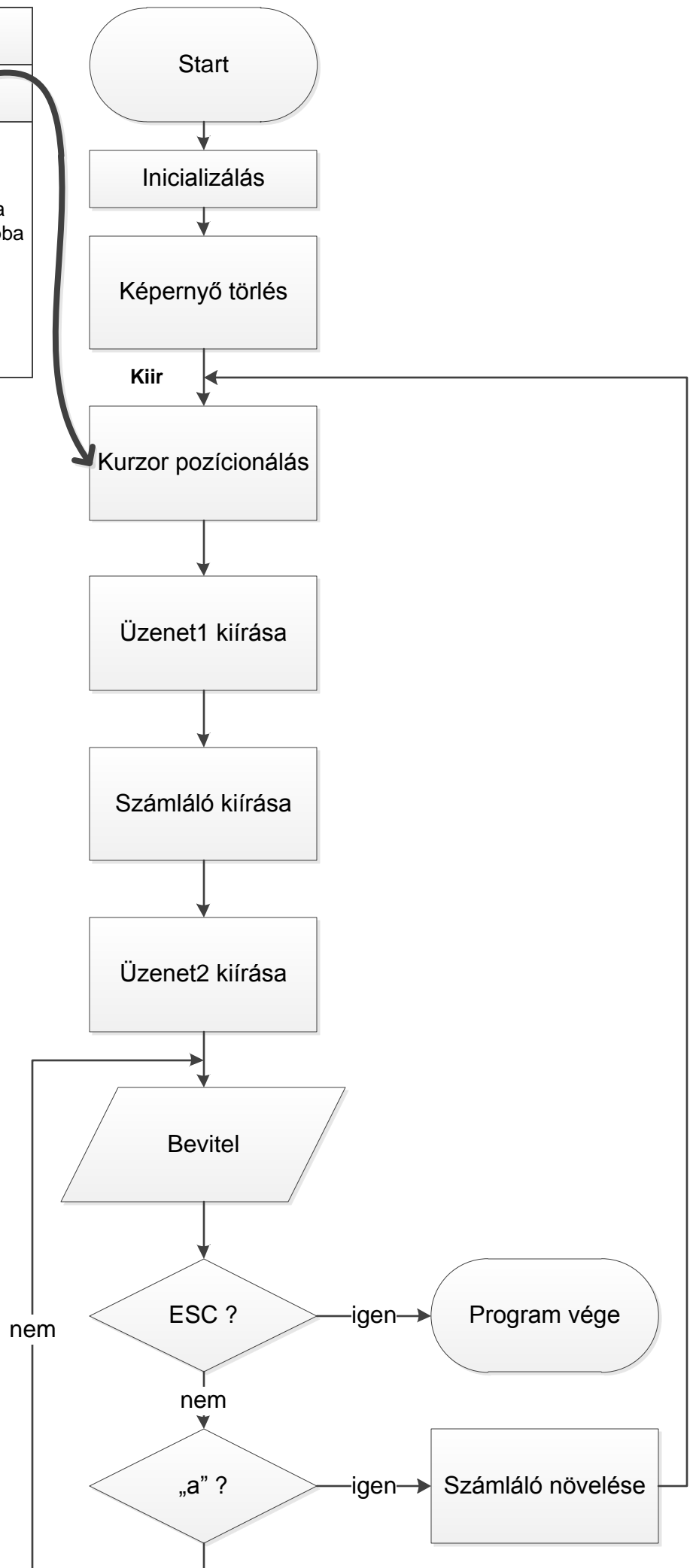
üzenet1: db "Az a billentyu \$"  
üzenet2: db " alkalommal volt leutve\$"  
szamlalo: db "0\$"



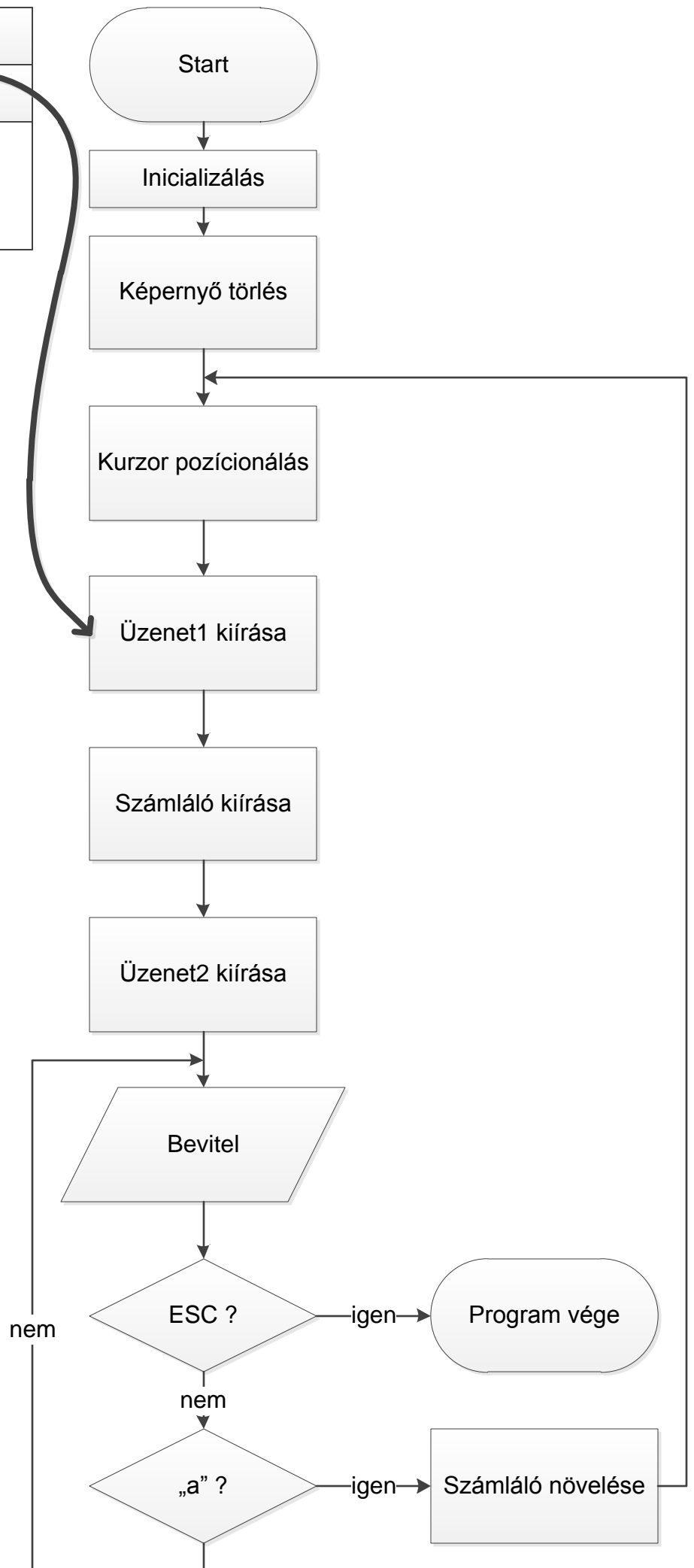
Implementálás	
	Képernyő törlés
mov	ax, 03h
int	10h



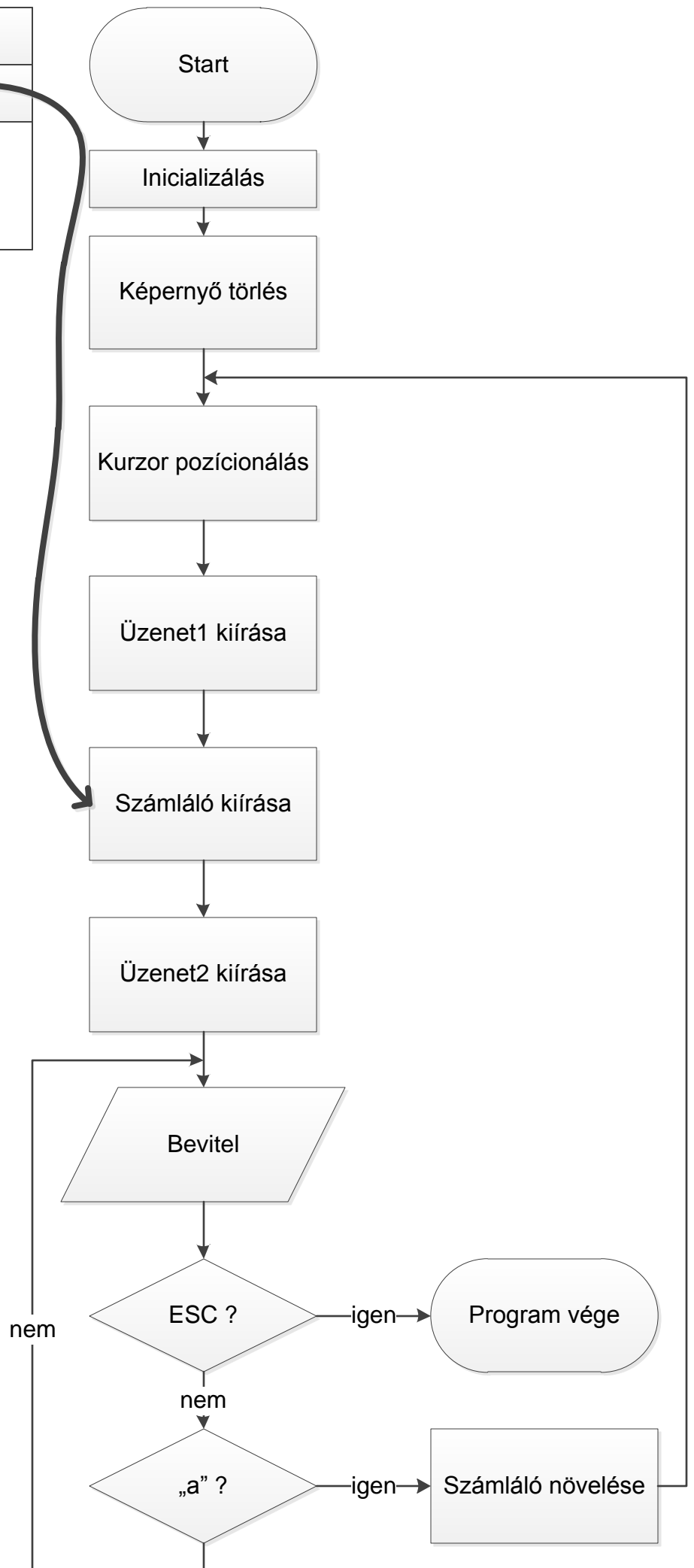
Implementálás	
Kurzor pozicionálás	
Kiir:	
mov	ah, 02h
mov	bh, 0 ;videó lap száma
mov	dh, 10 ;kurzor a 10. sorba
mov	dl, 0 ;kurzor a 0. oszlopba
int	10h



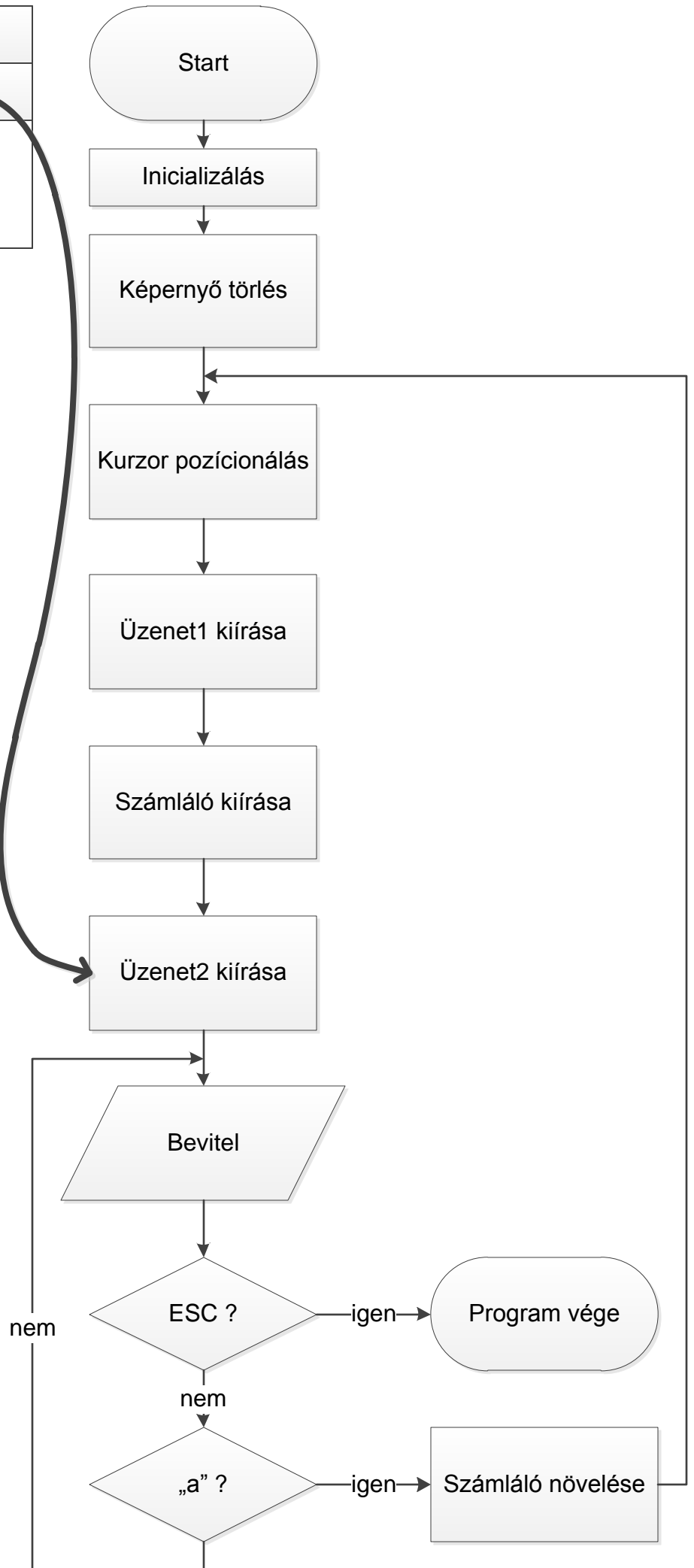
Implementálás	
	Üzenet1 kiírása
mov	dx, offset uzenet1
mov	ah, 09h
int	21h



Implementálás	
	Számláló kiírása
mov	dx, offset szamlalo
mov	ah, 09h
int	21h

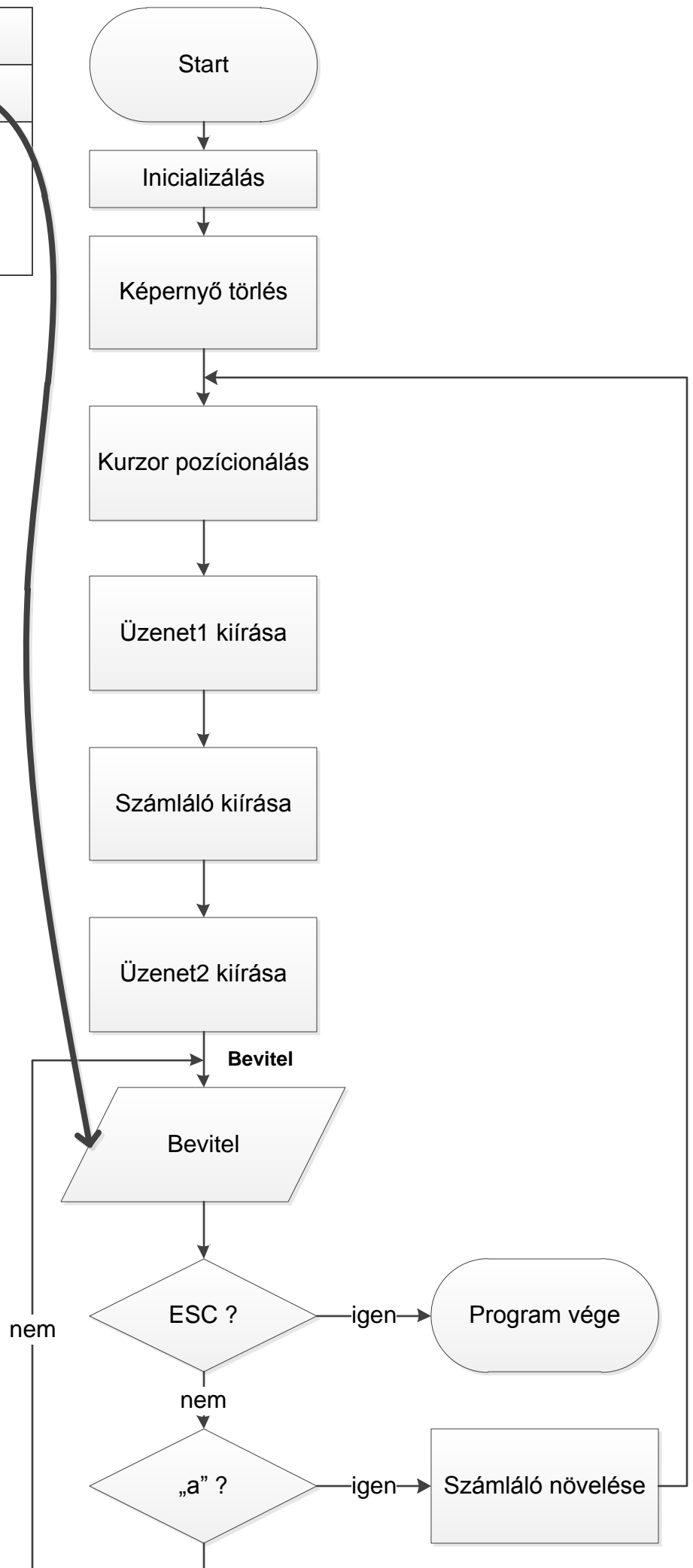


Implementálás	
	Üzenet2 kiírása
mov	dx, offset uzenet2
mov	ah, 09h
int	21h

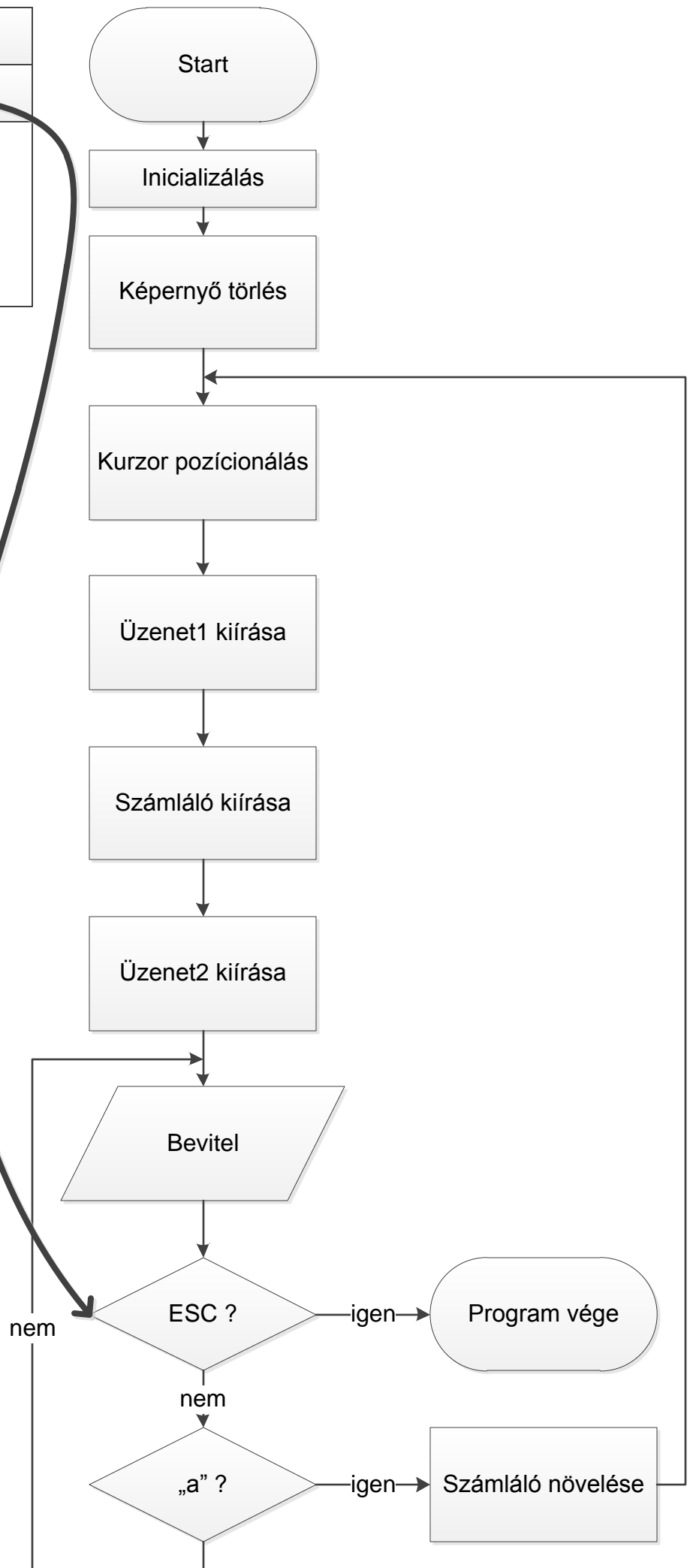




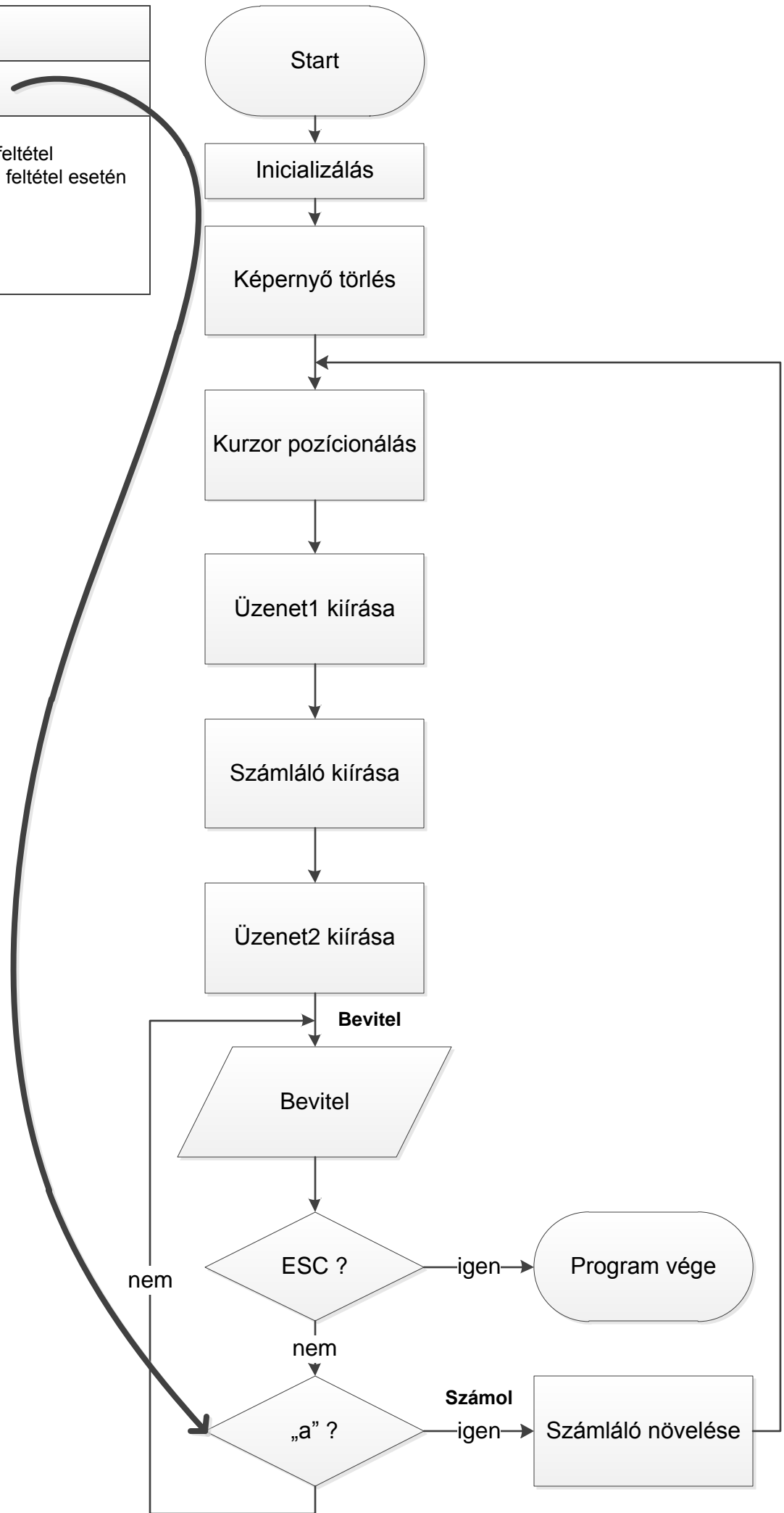
Implementálás	
	Bevitel
Bevitel:	
xor	ax, ax ;nullázás
int	16h ;várákozás egy
billentyű leütésre	



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



Implementálás	
	„a” ?
cmp	al, "a"
jz	Szamol ;igaz feltétel
jmp	Bevitel ;hamis feltétel esetén



Implementálás	
Számológép növelése	
Számol:	
mov	di, offset szamlalo
mov	al, [di]
inc	al
mov	[di], al
jmp	Kiir

