

26. LECKE /TÖBB LISTA HASZNÁLATA, MŰVELETEK LISTÁKKAL, ÖSSZEFÉSÜLÉS

27.) Logikai feladat:

Mennyi ideig isznak a matrózok?

Ha hat jókedvű, pörge bajszú, nagy pocakú matróz hat kupica pálinkát egy perc alatt tud legurítani, akkor 12 jókedvű, pörge bajszú, nagy pocakú matróz kétszer annyi kupica pálinkát mennyi idő alatt tud legurítani?



Megoldás: _____

Két rendezett tömböt, „a”-t és „b”-t összefésülünk egy harmadik „c” nevű tömbbe! Az a és b tömbök a program elején adottak, melyek elemeiből a program hozza létre a c tömböt.

a(1)	a(2)	a(3)	a(4)	a(5)	a(6)	a(7)	a(8)	b(1)	b(2)	b(3)	b(4)	b(5)	b(6)	b(7)	b(8)
4	7	9	10	16	17	21	30	5	12	13	18	20	22	24	25
c(1)	c(2)	c(3)	c(4)	c(5)	c(6)	c(7)	c(8)	c(9)	c(10)	c(11)	c(12)	c(13)	c(14)	c(15)	c(16)
4	5	7	9	10	12	13	16	17	18	20	21	22	24	25	30

A programban fogunk használni további két változót – ai és bi, melyek fogják jelölni, hogy az a ill. b tömb éppen melyik eleménél járunk.

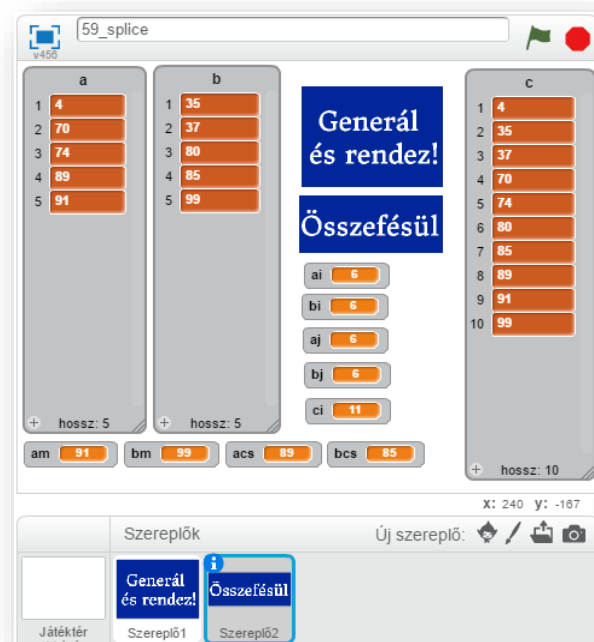
Az ai, bi változókat a program elején beállítjuk 1-re, majd a programban (a c tömb létrehozására szolgáló ciklusban) mindig az a tömb ai-dik eleme vagy a b tömb bi-dik eleme közül a kisebbet tesszük át a c tömbbe (és növeljük az ai vagy bi értékét egyel attól függően melyik tömbből raktuk át az elemet a c tömbbe). A c tömbhöz is bevezetünk egy ci változót, mely azt fogja jelölni, hogy éppen hol járunk a c tömbben (kezdetben ennek az értéke is 1 lesz, majd minden egyes átrakásnál növeljük egyel).

A fent leírt, elemek átrakására szolgáló algoritmust addig fogjuk ismételni, amíg nem érünk az a vagy a b tömb végére (ai vagy bi nem éri el a tömb végét, tehát amíg nem lesz több, mint az adott tömb elemeinek száma - an, bn). Ehhez egy „ismételd eddig” ciklust használunk. Ez után már csak a másik tömbből (amelyiknél még nem értünk a tömb végére) átrakjuk az összes megmaradt elemet a c tömbbe (ci-től kezdve a végéig).

32.) Gyakorlati feladat:

Nézzük meg gyakorlatban az „összefésülést”!
Gombnyomásra hozzunk létre két darab 5 elemű rendezett listát! A listák neve legyen „a” és „b”! Majd egy „c” nevű listába növekvő sorrendbe átmásoljuk megfelelő egymás utáni sorrendben a listák elemeit!
A feladat megoldásához nagyon sok változóra is szükségünk lesz!

- A program neve legyen: 59_splice!
- Hozzuk létre a 3 listát a megadott neveken, és készítsük el a két nyomógombot a minta szerint a képen látható felirattal!
- Az „a” lista rendezéséhez szükség van „ai”; „aj”; „am”; „acs”, változókra!
- Az „b” lista rendezéséhez szükség van „bi”; „bj”; „bm”; „bcs”, változókra!



- Egy „ci” változóra az összefésülésnél lesz szükségünk!
- A programot a zászlóra kattintással indítjuk, lenullázzuk az összes változót, és töröljük a listák tartalmát!
- A „Generál és rendez!” gombot úgy alakítjuk ki, hogy lenyomására generáljon mind a két listába 5-5 véletlen számot, 1 és 100 között, majd azonnal rendezze is minimumkiválasztásos rendezéssel mind a két tömböt!

```

ezen szereplőre kattintáskor
ismételd 5
  add hozzá: véletlen 1 és 100 között a listához
ismételd 5
  add hozzá: véletlen 1 és 100 között b listához
ismételd a hossza
  ai legyen ai + 1
  aj legyen ai + 1
  ismételd a hossza - ai
    am legyen ai eleme a
    ha am > aj eleme a akkor
      acs legyen aj eleme a
      cseréld le aj elemet a listában: ai eleme a
      cseréld le ai elemet a listában: acs
    aj legyen aj + 1
ismételd b hossza
  bi legyen bi + 1
  bj legyen bi + 1
  ismételd b hossza - bi
    bm legyen bi eleme b
    ha bm > bj eleme b akkor
      bcs legyen bj eleme b
      cseréld le bj elemet b listában: bi eleme b
      cseréld le bi elemet b listában: bcs
    bj legyen bj + 1
    
```

a	
1	4
2	70
3	74
4	89
5	91

```

-ra kattintáskor
ai legyen 0
bi legyen 0
ci legyen 0
aj legyen 0
bj legyen 0
am legyen 0
bm legyen 0
acs legyen 0
bcs legyen 0
töröld minden elemet a listából
töröld minden elemet b listából
töröld minden elemet c listából
    
```

```

ezen szereplőre kattintáskor
ai legyen 1
bi legyen 1
ci legyen 1
ismételd eddig: ai > a hossza vagy bi > b hossza
  ha ai eleme a < bi eleme b akkor
    szűrd be: ai eleme a ci helyen c listába
    ai legyen ai + 1
    ci legyen ci + 1
  különben
    szűrd be: bi eleme b ci helyen c listába
    bi legyen bi + 1
    ci legyen ci + 1
  ha ai > a hossza akkor
    ismételd b hossza - bi + 1
      szűrd be: bi eleme b ci helyen c listába
      bi legyen bi + 1
      ci legyen ci + 1
    különben
      ismételd a hossza - ai + 1
        szűrd be: ai eleme a ci helyen c listába
        ai legyen ai + 1
        ci legyen ci + 1
    
```

b	
1	35
2	37
3	80
4	85
5	99

c	
1	4
2	35
3	37
4	70
5	74
6	80
7	85
8	89
9	91
10	99

- Az elkészített programon tudatosan soronként végigmenve nézzük meg, hogy mit miért csinálunk! A változók értékekkel való helyettesítéssel menjünk végig a ciklusokon, és értelmezzük a parancsokat!
- Az összefésülést bontsuk két részre, az elsőben addig haladunk, amíg elérünk az egyik tömb utolsó eleméhez! Az első részben így már sorba vannak rendezve az eddigi elemek a „c” tömbben!
- A másik lista elemeivel már csak annyi dolgunk van, hogy hozzátegyük „c” tömb végéhez, mert azok már eleve rendezve vannak!