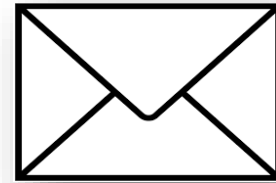


24. LECKE / EGY LISTA ELEMEINEK NÖVEKVŐ SORRENBE TÉTELE, MINIMUMKIVÁLASZTÁSOS RENDEZÉSEL

25.) Logikai feladat:

Két borítékot és két képeslapot vettem. Egy boríték 5 Ft-tal olcsóbb, mint egy képeslap. 50 Ft-ot fizettem. Mennyibe került egy képeslap, mennyibe egy boríték?



Képeslap: _____ Ft Boríték: _____ Ft

Az előző módszer hátránya a sok felesleges csere. Célszerűbb lenne az aktuális, i , elemet a mögötte lévők közül egyedül a legkisebbel felcserélni. Ez a felismerés vezet a módszer javításához, a minimumkiválasztásos rendezéshez.

A minimumkiválasztásos rendezés lépései:

Az $a()$ tömb 1. helyére keressük ki az $a(1)$, $a(2)$, $a(3)$, $a(4)$, $a(5)$ elemek közül a legkisebbet (minimumkiválasztás)! Ha megtaláltuk a legkisebb elemet, akkor cseréljük ki az $a(1)$ -gyel!

Ekkor az $a(1)$ biztosan a vektor legkisebb eleme. Növeljük az $i - 1$ -gyel, így az $a()$ tömb 2. helyére keressük ki $a(2)$, $a(3)$, $a(4)$, $a(5)$ közül a legkisebbet!

Most $a(i)$ -t cseréljük fel $a(\min)$ -nel! (Ez a csere tulajdonképpen felesleges, hiszen most $a(2)$ -t cseréljük fel $a(2)$ -vel! növeljük $i - 1$ -gyel!

Most $a(i)$ -t cseréljük fel $a(\min)$ -nel! majd növeljük $i - 1$ -gyel!

Most $a(i)$ -t cseréljük fel $a(\min)$ -nel! az i -vel elértük $n-1$ -t, ezután az $a()$ biztosan rendezett!

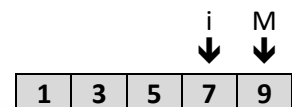
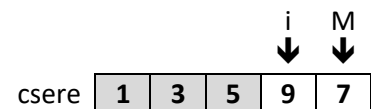
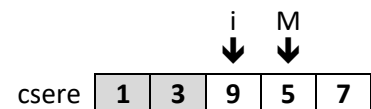
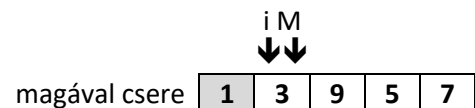
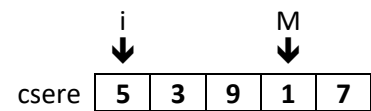
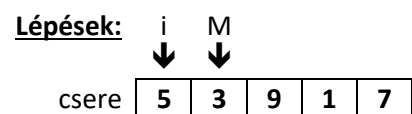
A módszer hátránya, hogy bizonyos esetekben (ha $i = \min$) egy elemet önmagával cserélünk fel. Ez nyilvánvalóan felesleges csere.

Alap:

	1	2	3	4	5
rendezetlen	5	3	9	1	7

Végeredmény:

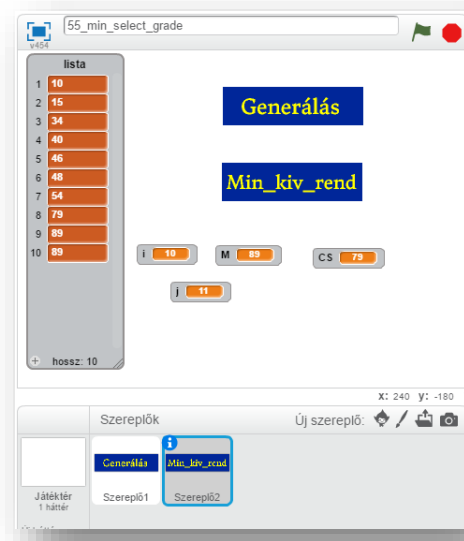
	1	2	3	4	5
rendezett	1	3	5	7	9



30.) Gyakorlat:

Készítsünk az előző algoritmus felhasználásával egy programot, melyben egy 10 elemű tömb elemeit rendezzük növekvő sorrendbe minimumkiválasztásos rendezéssel!

- A program neve legyen: 55_min_select_grade!
- A lista neve legyen: lista! A programhoz szükséges változók legyenek: i ; j ; M (minimum); CS (csere)!
- Készíts két szereplőt gombnak a minta szerint!
- Futtasd, és teszteld a programot, majd mentsd a megadott helyre!



```

-ra kattintáskor
i legyen 0
j legyen 0
M legyen 0
CS legyen 0
töröld minden elemet lista listából

ezen szereplőre kattintáskor
ismételd 10
add hozzá: véletlen 1 és 100 között lista listához

ezen szereplőre kattintáskor
ismételd lista hossza
i legyen i + 1
j legyen i + 1
ismételd lista hossza - i
M legyen i eleme: lista
ha M > j eleme: lista akkor
CS legyen j eleme: lista
cseréld le j elemet lista listában: i eleme: lista
cseréld le i elemet lista listában: CS
j legyen j + 1
    
```

26.) Önálló feladat:

Ebben a programban egy „taco-evő verseny” eredményit kell ábrázolni! A versenyen 10 ember indult, és 10 tacónál senki nem tud többet enni!

- A program neve legyen 56_taco!
- Először generálj egy listába véletlenszerűen számokat az előzőekben leírtak szerint!
- Majd a listában szereplő számokat rendezd minimumkiválasztásos rendezéssel csökkenő sorrendbe!
- Az eddigieket, mind egy gomb lenyomására tegye meg!
- A rendezett lista alapján rajzolja ki, hogy az első helyezett hány darab taco-t tudott megenni!
- Majd alá, a második, harmadik, és így tovább a többi helyezett teljesítményét!
- A programot a mintakép alapján készítsd el, de a saját elképzelésed szerint programozd le!
- A lista, a változók, a szereplők nevének megadásában is szabadkezedet kapsz!
- A program futtatása alatt a lista és a változók legyenek elrejtve!
- Futtasd, és teszteld a programot, majd mentsd a megadott helyre!

