Belépő a tudás közösségébe

Informatika szakköri segédanyag

Programozási tételek összeépítése

Heizlerné Bakonyi Viktória, Horváth Győző, Menyhárt László, Szlávi Péter, Törley Gábor, Zsakó László

Szerkesztő: Abonyi-Tóth Andor, Zsakó László

A kiadvány „A felsőoktatásba bekerülést elősegítő készségfejlesztő és kommunikációs programok megvalósítása, valamint az MTMI szakok népszerűsítése a felsőoktatásban” (EFOP-3.4.4-16-2017-006) című pályázat keretében készült 2017-ben.

Eötvös Loránd Tudományegyetem
Informatikai Kar

SZÉCHENYI 2020
MAGYARORSZÁG KORMÁNYA
Európai Unió
Európai Szociális Alap
BEFEKTETÉS A JÖVŐBE

Gyakran előfordul, hogy programozási tételeket egymás után kell használnunk. Ezen egymásutániságnál azonban sok esetben a két megoldó algoritmus egybeépítése egyszerűbb, rövidebb, hatékonyabb megoldást eredményez. Ebben a részben ezekkel foglalkozunk. Az egymásra építés mindig két programozási tétel összefogását jelenti, a fejezeteket a korábban alkalmazandó tétel szerint fogalmaztuk meg.

1. Másolással összeépítés

Másolással bármelyik programozási tétel egybeépíthető, hiszen csupán annyi a teendő, hogy a programozási tételben szereplő `X[i]` bemenő adatra hivatkozást kicseréljük `g(X[i])`-re.

Ha például egy számsorozat elemeinek négyzetösszegét kellene megadnunk, az egy másolást (számokhoz számok négyzetei rendelése) és egy összegzést tartalmaz. Nézzük meg e két tétel általános egymásra építését! A megoldásban – mint azt az összegzésnél tettük – induljunk ki a nullelemből, alkalmazzuk a `f` függvényt az addig kiszámított értékre és a sorozat eleméből kiszámított értékre!

```
Másolás_összegzés(N,X,S):
  S:=0
  Ciklus                          I=1-től                          N-ig
    S:=S+X[I]*X[I]
  Ciklus                                                           vége
Eljárás vége.
```

Második példaként vegyük a másolás és a maximumkiválasztás összeépítését! Ebben a maximális elem értékét és az indexét is meghatározzuk. Az előző feladat analógiájára ilyen lehet a **legnagyobb abszolút értékű** szám abszolút értékének meghatározása egy számsorozatból.

Ebben a megoldásban – a maximumkiválasztás alapján – vegyük az első elemből kiszámított függvényértéket induló maximumnak, ezt hasonlítsuk a további elemekből kiszámított függvényértékekkel, és a legnagyobbat őrizzük meg!

```
Másolás_maximumkiválasztás(N,X,Max,Maxert):
  Max:=1;                                                  Maxert:=|X[1]|
  Ciklus                          i=2-től                          N-ig
    Ha     Maxert<|X[i]|     akkor     Maxert:=|X[i]|;     Max:=i
  Ciklus                                                           vége
Eljárás vége.
```

Minta kódok

C++	cpp/1_Masolas_Maximumkivalasztas/feladat.cpp
C#	cs/1_Masolas_Maximumkivalasztas/feladat.cs
Java	java/1_Masolas_Maximumkivalasztas/feladat.java
Pascal	pas/1_Masolas_Maximumkivalasztas/feladat.pas
Python	py/1_Masolas_Maximumkivalasztas/feladat.py

2. Megszámolással összeépítés

A megszámolást három elemi programozási tétellel érdemes egybeépíteni, az eldöntéssel, a kiválasztással, valamint a kereséssel.

Itt olyan kérdéseket tehetünk fel, hogy van-e egy sorozatban legalább `K` db `T` tulajdonságú elem, adjuk meg a sorozat `K`-adik `T` tulajdonságú elemét stb. Egy elem legyen pl. `T` tulajdonságú, ha pozitív!

Az általánossága miatt nézzük a megszámolás és a keresés egymásra építését! Az eldöntésnél, illetve a kiválasztásnál hasonlóan kellene eljárnunk, hiszen e két típusalgoritmus megoldásszövege része a keresés megoldásszövegének.

Induljunk ki a keresés megoldásából! A keresés ciklusa akkor állt le, amikor megtaláltuk az első `T` tulajdonságú elemet. Ezt kell kicserélni arra, hogy csak a `K`-adiknál álljon le (ha egyáltalán van `K`). A ciklusmagban viszont számolnunk kell a `T` tulajdonságú elemeket – ahogyan azt a megszámolásnál tettük!

```
Megszámolás_Keresés(N,X,K,Van,Sorsz):
  i:=0; db:=0
  Ciklus amíg i<N és db<K
    i:=i+1
    Ha X[I]>0 akkor db:=db+1
  Ciklus vége
  Van:=(db=K)
  Ha Van akkor Sorsz:=i
Eljárás vége.
```

Minta kódok

C++	cpp/2_Megszamolas_Kereses/feladat.cpp
C#	cs/2_Megszamolas_Kereses/feladat.cs
Java	java/2_Megszamolas_Kereses/feladat.java
Pascal	pas/2_Megszamolas_Kereses/feladat.pas
Python	py/2_Megszamolas_Kereses/feladat.py

3. Maximumkiválasztással összeépítés

Maximumkiválasztással kapcsolatban azt a kérdést fogalmazhatjuk meg, hogy hány darab maximális értékű elem van, s hogy melyek a maximális értékű elemek.

Itt tehát a maximumkiválasztást kell egybeépíteni a megszámolással, illetve a kiválogatással.

Az a kérdés, hogy van-e egyáltalán több maximális értékű elem, egy menetben nem dönthető el, azaz a három tételt nem lehet egymásba építeni, hanem csak egymás után alkalmazni.

A korábbiakban megállapítottuk, hogy a kigyűjtéses kiválogatás mindig tartalmaz egy megszámolást, így csak a kiválogatással kell foglalkoznunk.

Az összeépítés alapgondolata, hogy a „lokális" maximumok megőrzésével együtt válogassuk is ki a lokális maximumokat. Természetesen új lokális maximum megtalálásakor a korábbi kigyűjtést el kell felejteni. A kiválasztó ciklus végén a lokális maximum éppen a keresett maximumérték, tehát az éppen kigyűjtött sorszámok a maximumok sorszámai lesznek.

```
Maximumkiválogatás(N,X,Db,Y,Maxert):
  Maxert:=X[1]; Db:=1; Y[Db]:=1
  Ciklus i=2-től N-ig
    Elágazás
      X[i]>Maxert  esetén Maxert:=X[i]; Db:=1; Y[Db]:=i
      X[i]=Maxert  esetén Db:=Db+1; Y[Db]:=i
    Elágazás vége
  Ciklus vége
Eljárás vége.
```

Minta kódok

C++	cpp/3_Maximumkivalogatas/feladat.cpp
C#	cs/3_Maximumkivalogatas/feladat.cs
Java	java/3_Maximumkivalogatas/feladat.java
Pascal	pas/3_Maximumkivalogatas/feladat.pas
Python	py/3_Maximumkivalogatas/feladat.py

Feladatok programozási tételekre a Nemes Tihamér OITV-ről és az Informatika OKTV-ről

1. feladat

Egy lövészversenyen a versenyzők egymás után lőnek. Ismerjük N ($1 \leq N \leq 1000$) versenyző eredményét. Készíts programot, amely beolvassa N értékét és az N darab eredményt, majd megadja:

A. minden versenyzőre, hogy az addig szereplők közül hányan értek el nála jobb eredményt;

B. azokat a versenyzőket, akik a verseny valamelyik időszakában álltak az első helyen;

C. azokat a versenyzőket, akik a verseny valamelyik időszakában álltak az utolsó helyen;

D. a verseny győzteseit!

Példa

```
Bemenet:                Kimenet:
N=6                     Jobb eredmény: 0 0 2 0 3 3
1. versenyző: 594       Állt az első helyen: 1 2 4
2. versenyző: 596       Állt az utolsó helyen: 1 3
3. versenyző: 582       Győztesek: 4
4. versenyző: 599
5. versenyző: 590
6. versenyző: 590
```

A. nem összeépítés

B. az összes maximum számításakor egy új maximális érték megtalálása esetén ne töröljük a korábbi maximumokat;

C. az összes minimum számításakor egy új maximális érték megtalálása esetén ne töröljük a korábbi minimumokat;

D. az összes maximum számításakor egy új maximális érték megtalálása esetén töröljük a korábbi maximumokat!

```
B(Bdb,Bt):
  max:=1; Bdb:=1; Bt[1]:=1
  Ciklus i=2-től n-ig
    Ha pont[i]≥pont[max] akkor Bdb:=Bdb+1; Bt[Bdb]:=i
    Ha pont[i]>pont[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

```
C(Cdb,Ct):
  max:=1; Cdb:=1; Ct[1]:=1
  Ciklus i=2-től n-ig
    Ha pont[i]≤pont[max] akkor Cdb:=Cdb+1; Ct[Cdb]:=i
    Ha pont[i]<pont[max] akkor max:=i
  Ciklus vége
Eljárás vége.

D(Ddb,Dt):
  max:=1; Ddb:=1; Dt[1]:=1
  Ciklus i=2-től n-ig
    Ha pont[i]=pont[max] akkor max:=i; Ddb:=1; Dt[1]:=i
    különben ha pont[i]>pont[max] akkor Ddb:=Ddb+1; Dt[Ddb]:=i
  Ciklus vége
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	101. Lövészverseny eredményei

2. feladat – G(x)-ek összege

Egy kártyajátékban az egyes lapoknak számértékük van. Minden lapot egy színnel és egy figurával adunk meg. A színek: piros, zöld, tök, makk. A figurák: 7-es, 8-as, 9-es, 10-es, alsó, felső, király, ász. A számot tartalmazó figurák annyit érnek, amennyi a ráírt szám. Az alsó 2-t, a felső 3-at, a király 4-et, az ász 11-et ér. A piros lapoknál az értéket duplán kell számítani.

Készíts programot, amely beolvassa egy játékos N (1≤N≤4) kártyáját, majd megadja, hogy a lapok összesen hány pontot érnek!

Példa

```
Bemenet:
Kártyák száma? 3
1. kártya színe? piros
1. kártya figurája? alsó
2. kártya színe? tök
2. kártya figurája? 7-es
3. kártya színe? tök
3. kártya figurája? ász
```

```
Kimenet:
A kártyák értéke: 22
```

Tároljuk a `Szin` vektorban a lehetséges kártyaszíneket, a pirossal kezdve; a `Figura` vektorban pedig a kártyafigurákat úgy, hogy az értékük éppen az indexük legyen (üresen hagyva azokat az indexű elemeket, amilyen értékű figura nincs)!

A feladat megoldása a lapok értékeinek összegzése, ahol az értéket egy speciális függvény-nyel számítjuk ki:

```
Kártya(N,Szin,Figura,E):
  E:=0
  Ciklus i=1-től N-ig
    j:=1
    Ciklus amíg s[i]≠Szin[j]
      j:=j+1
    Ciklus vége
    Ha j=1 akkor sz:=2 különben sz:=1
    j:=1
    Ciklus amíg f[i]≠Figura[j]
      j:=j+1
    Ciklus vége
    E:=E+sz*j
  Ciklus vége
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	100. Kártyaleosztás értéke

3. feladat

Van N darab egységnyi méretű négyzetlapunk. K×K-as négyzeteket kell összerakni belőlük, először a lehető legnagyobbat, utána a maradékból egyre kisebbeket, ...

Írj programot, amely beolvassa a négyzetlapok számát ($1 \leq N \leq 10000$), majd megadja, hogy a fenti elven mekkora négyzetek rakhatók ki belőlük!

Példa

`N=72 ⇒ 8×8-as négyzet, 2×2-es négyzet, 2×2-es négyzet`

A feladat megoldása speciális függvénnyel számolt értékek összegzése.

Meg kell találni az `N` előtti legnagyobb négyzetszámot, majd ezt levonva `N`-ből újrakezdeni ezt az eljárást. Ezt mindaddig végezzük, amíg `N`-re `0`-t nem kapunk.

```
Négyzet(N):
  k:=1
  Ciklus amíg (k+1)*(k+1)≤N
    k:=k+1
  Ciklus vége
  Ciklus amíg N>0
    Ki: k; N:=N-k*k
    Ciklus amíg k*k>N
      k:=k-1
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	102. Négyzetek összerakása

4. feladat

Ismerjük az A ($1 \leq A \leq 32767$) pozitív egész számot, valamint azt is tudjuk, hogy az A számjegyeinek összege valamilyen számrendszerben éppen S. A számrendszer alapszáma biztosan nem nagyobb A+1-nél.

Készíts programot, amely beolvassa az adatokat, majd kiírja a képernyőre, hogy az A szám számjegyeinek összege mely számrendszerekben lehet éppen S!

Példa

```
Bemenet:  A=127, S=3

Kimenet: Lehetséges számrendszer=5, 63, 125

Magyarázat: 127=1*5^3+0*5^2+0*5+2, 127=2*63+1, 127=1*125+2
```

A megoldás egy speciális elemekből álló összeg kiszámolása, ahol az elemeket magukat is számoljuk. Egy `X` szám számjegyei összegét `A` alapú számrendszerben a következőképpen számoljuk ki:

```
Összeg(X,A):
  b:=0
  Ciklus amíg X>0
    b:=b+X mod A; X:=X div A
  Ciklus vége
  Összeg:=b
Függvény vége.
```

Ezután a feladat megoldása:

```
Számrendszer(A,S):
  Ciklus i=2-től A+1-ig
    Ha Összeg(A,i)=S akkor Ki: i
  Ciklus vége
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Számelméleti algoritmusok
Feladat	31. Számrendszer

5. feladat

Egy nemzetközi vonat több napon keresztül megy az egyik végállomásáról a másik végállomására. Ismerjük minden állomásra az érkezési időt. A vonat a végállomás kivételével minden állomáson pontosan 10 percet várakozik, majd tovább indul.

Készíts programot, amely beolvassa az állomások számát ($2 \leq N \leq 100$), a kezdő állomásról indulási időt ($0 \leq$ óra ≤ 23, $0 \leq$ perc ≤ 59), majd pedig a további N-1 állomásra érkezési időt (óra, perc). A program ezekből számítsa ki, hogy

A. hány perc volt a leghosszabb időszak, amikor a vonat sehol sem állt meg;

B. a vonat mely állomások között haladt (vagy mely állomásokon állt) éjfélkor!

Példa

```
Bemenet: N=7; indulás=9 óra 20 perc; érkezések: (13,30),
(19,45), (4,00), (16,30), (23,55), (6,30).
Leghosszabb menetidő: 740 perc {a 4. és az 5. állomás között}
```

A. speciális értékek maximumát kell meghatározni (a menetidőbe a várakozási időt nem szabad beszámítani);

B. nem tétel összeépítés.

Érdemes az indulási és érkezési időket percre átszámítani (`Idő(i)`). Ekkor a szomszédos értékek különbsége (figyelembe véve a `10` perces várakozást) maximuma az első részfeladat megoldása.

A második részfeladatban azok a vonatok állnak éjfélkor valamelyik állomáson, amelyek éjfél előtt legfeljebb 10 perccel érkeztek, s azok haladnak két állomás között, amelyek az egyik állomáson még éjfél előtt voltak, a másikra pedig éjfél után érkeztek.

```
VonatokA(N,Idő,Legh):
  Legh:=Idő(2)-Idő(1)
  Ciklus i=3-tól N-ig
    Ha Idő(i)>Idő(i-1)   {ugyanazon A napon vannak-e?}
       akkor Ha Idő(i)-Idő(i-1)-10>Legh
                akkor Legh:=Idő(i)-Idő(i-1)-10
    különben Ha Idő(i)+24*60-Idő(i-1)-10>Legh
                akkor Legh:=Idő(i)+24*60-Idő(i-1)-10
  Ciklus vége
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	97. Nemzetközi vonat állomásai

6. feladat

Egy hegymászó a tervezett útvonala mentén méterenként megmérte a felszín tengerszint feletti magasságát. N helyen kapott mérési adatokat. Emelkedőnek nevezzük azt a számsorozatot, amelynek minden eleme nagyobb, mint az előtte levő. Az emelkedő helye az ilyen számsorozat első és utolsó tagjának sorszáma, a hossza pedig a számsorozatban levő számok darabszáma. (Emelkedő lehet balról jobbra, illetve jobbról balra haladva is!)

Készíts programot, amely megadja, hogy az út során hol volt a leghosszabb emelkedő! (Ha több egyforma van, közülük egyet kell megadni.) Ha nincs emelkedő az út során, akkor írja ki, hogy NINCS!

A feladat megoldásához először be kell olvasni a mérések számát ($1 \leq N \leq 100$), majd pedig az N darab mérést; majd ezután ki kell írni az eredményt.

Példa

```
Bemenet: N=10
         Mérések: 100,110,115,110,105,115,125,130,125,125
Kimenet: 5,8
```

A feladat megoldásához meg kell határozni az oda- és a visszaúton is az emelkedők kezdetét és végét, s közben közülük ki kell választani a leghosszabbat. Ha nincs emelkedő, akkor a leghosszabb emelkedő kezdete `0` marad.

```
Leghosszabb_emelkedő(N,H,Nincs, Maxk,Maxv):
  Maxk:=0; Maxv:=0; k:=0
  Ciklus i=2-től N-ig
    Ha k=0 és H[i]>H[i-1] akkor k:=i-1
    Ha k>0 és H[i]≤H[i-1] akkor
       Ha i-k>Maxv-Maxk+1 akkor Maxk:=k; Maxv:=i-1
       k:=0
    Elágazás vége
  Ciklus vége
  Ha k>0 akkor Ha N+1-k>Maxv-Maxk+1 akkor Maxk:=k; Maxv:=N
  k:=0
  Ciklus i=N-1-től 1-ig
    Ha k=0 és H[i]>H[i+1] akkor k:=i+1
    Ha k>0 és H[i]≤H[i+1] akkor
      Ha k-i>|Maxk-Maxv|+1 akkor Maxk:=k; Maxv:=i+1
      k:=0
    Elágazás vége
  Ciklus vége
  Ha k>0 akkor Ha k>|Maxk-Maxv|+1 akkor Maxk:=k; Maxv:=1
  Nincs:=(Maxk=0)
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	30. Leghosszabb emelkedő

7. feladat

A Villamos-közlekedési Vállalat (VKV) felmérést végzett a villamosok kihasználásáról, melyet számítógéppel kell feldolgozni. A villamos-vonalon N állomás van, beleértve az induló- és a végállomást is. Egy út során a villamosvezetőnek meg kellett számolnia minden állomáson a fel- és a leszállókat, s neked ezekből az adatokból kell adott jellemzőket kiszámolnod.

Készíts programot, amely beolvassa N ($2 \le N \le 100$) értékét, a vezető által adott számokat (N*2 adat, mindegyik pozitív), majd belőlük a következőket határozza meg, és írja ki a képernyőre:

A. Hány ember utazott összesen a villamoson?

B. Mely állomásokon szállt le a villamosról az összes utas?

C. Mi volt a villamoson a maximális utasszám?

D. Hány állomásközi szakaszt tett meg a villamos úgy, hogy egyetlen utas sem volt rajta?

Példa

```
Bemenet:

  5 állomás

  Felszállók: 5 3 0 2 0

  Leszállók:  0 4 4 0 2

A: Összesen 10 ember utazott a villamoson.

B: Mindenki leszállt a 3. és az 5. állomáson.

C: A maximális utasszám 5 volt.

D: 1 szakaszon nem volt utas.
```

Jelölje `Fel[i]` az `i`-edik állomáson felszállók, `Le[i]` pedig a leszállók számát! Az egyik részfeladat a tartalmi helyesség ellenőrzése.

A. Nem tételösszeépítés – egyszerűen a felszállók számának összege.

B. Jelölje az `utasszam[i]` az utasok számát az `i`-edik állomásra érkezve, leszállás után:

$$\text{utasszam[i]} = \sum_{j=1}^{i-1} \text{Fel[j]} - \sum_{j=1}^{i} \text{Le[j]}$$

Ekkor a feladat az `utasszam[i]=0` értékek megtalálása.

C. A maximális `Utasszam[i]+Fel[i]` meghatározása a feladat.

D. Az `utasszam[i]+Fel[i]=0` értékek száma a feladat.

Másodikként ki kell válogatni azon állomások sorszámát, amikor a leszállás után (a felszállás előtt) az utasszám `0` lett (`Db`, `Hely`). Ehhez természetesen számolni kell a pillanatnyi utasszámot (`utasszam`).[1] Ez utóbbi maximuma lesz a C részfeladat megoldása (`Max`). A D részfeladat megoldásához azon esetek számát kell megadni, amikor a felszállások után `0` volt az aktuális utasszám.

[1] Nincs szükség az utasszám vektorra, mert mindig elegendő a pillanatnyi.

```
Villamos(N,Fel,Le,Max,Db,Hely,Uresek):
  utasszam:=0; Max:=0; Db:=0; Uresek:=0
  Ciklus i=1-től N-ig
    utasszam:=utasszam-Le[i]
    Ha utasszam=0 és i>1 és Le[i]≠0
                    akkor Db:=Db+1; Hely[Db]:=i
    utasszam:=utasszam+Fel[i]
    Ha utasszam>Max akkor Max:=utasszam
    Ha utasszam=0 és i<n akkor Uresek:=Uresek+1
  Ciklus vége
Eljárás vége.
```

Az A) feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek: sorozatszámítás
Feladat	9. Összes utas száma a villamoson *

A B) feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	104. Villamos statisztika

A C) feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek: minimum, maximum számítás
Feladat	31. Maximális utasszám a villamoson *

A D) feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek: megszámolás
Feladat	46. Utas nélküli állomásszakaszok száma *

8. feladat

Ádám és Éva N játékot játszott egymással. Minden játékról tudjuk, hogy melyikük nyerte meg.

Készíts programot, amely megadja, hogy Ádám vagy Éva nyert-e többször! (Páros N esetén elképzelhető, hogy egyik sem).

Példa

```
Bemenet:              Kimenet:

5                     Ádám – hamis
Ádám                  Éva – igaz
Éva
Éva
Éva
Ádám
```

A feladat átfogalmazva: nyert-e Éva legalább N/2-ször, vagy nyert-e Ádám legalább N/2-ször. Ebből következően nem kell feltétlenül az összes eredményt végig néznünk.

```
Ádám_Éva(A,E):
  A:=hamis; E:=hamis; i:=0; db:=0
  Ciklus amíg i<N és db≤N/2 és i-db≤N/2
    i:=i+1
    Ha X[i]='Ádám' akkor db:=db+1
  Ciklus vége
  A:=db>N/2; E:=(i-db)>N/2
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Programozási tételek összeépítése
Feladat	96. Ádám vagy Éva nyert többször?