

1. Feladat: Szöveg „tömörítése” (szóközöket eltávolítjuk, s minden szót nagybetűvel kezdünk)

Példa bemenet	Példa kimenet
A tanévben a legjobbak a szünetek.	ATanévbenALegjobbakASzünetek.

Megoldási gondolatmenet:

Beolvasom a szöveget egy változóba, majd egyesével végigmegeyek a karaktereken, s ha szóközt találok, azt nem másolom át, valamint ha szóköz után jön kisbetű, akkor a neki megfelelő nagybetűt másolom át.

Megoldási algoritmus:

Változók:

sz,k: szöveg;  
i: egész szám;

BE: sz;  
k:="";

HA sz[1]≠' ' AKKOR  
k:=sz[1];  
ELÁGAZÁS VÉGE

CIKLUS i:=2-től hossz(sz)-ig  
HA sz[i] ≠' ' AKKOR  
HA (sz[i-1]≥'a') ÉS (sz[i-1]≤'z')  
AKKOR  
k:=k+nagybetűs(sz[i]);  
KÜLÖNBEN  
k:=k+sz[i];  
ELÁGAZÁS VÉGE  
ELÁGAZÁS VÉGE  
CIKLUS VÉGE  
KI: k;

Megoldási algoritmus magyarázata:

sz változóban tároljuk a beolvasott szöveget,  
k változóban tároljuk a kódolt szöveget  
i ciklusváltozóval egyesével vizsgáljuk a szöveg betűit

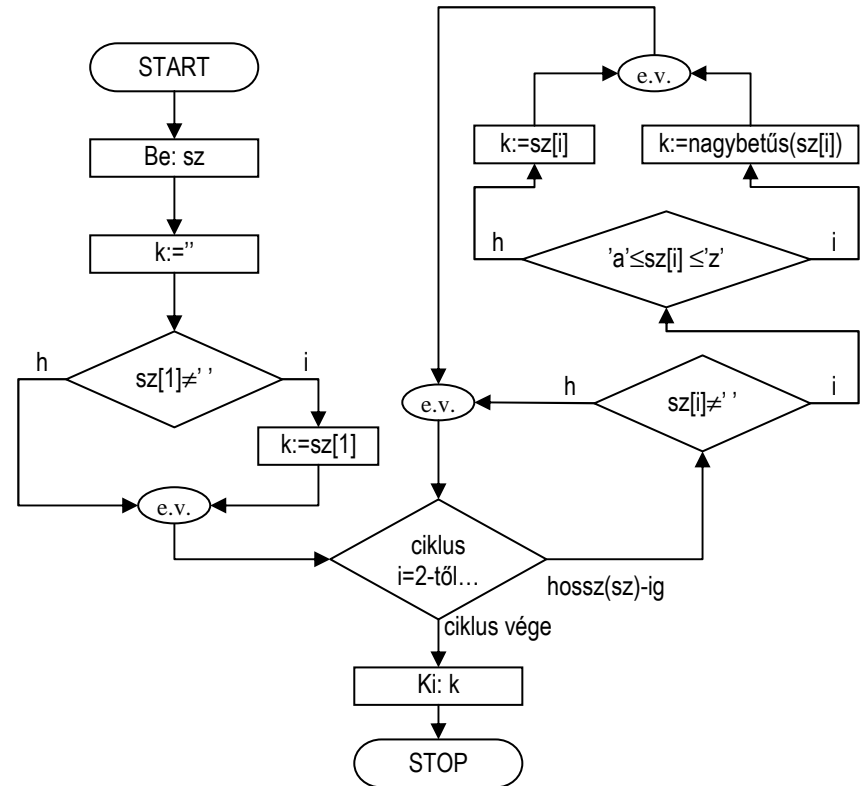
beolvassuk a szöveget a felhasználótól  
„lenullázzuk” a k változót

megnézzük külön az első karaktert, s ha nem szóköz,  
akkor átmásoljuk (feltesszük, hogy nagy kezdőbetűvel  
írták a szöveget)

a 2.-tól indítjuk a vizsgálatot, mert vizsgálni kell az  
előzőt is, (s ha 1-től kezdenénk, akkor a 0.-at is  
vizsgálni kellene)  
csak akkor másolom, ha nem szóköz, de  
ha kisbetű, vagyis 'a'...'z' közti betű  
akkor a nagybetűs párját másolom át  
különb  
magát az i. karaktert.

kíratom a „kódolt” szöveget

Folyamatábra



2. Feladat: Szöveg „szendvics” kódolása (első körben fel fogjuk használni az 1.feladatot, hogy tömörítsük a szöveget)

Példa bemenet		Példa kimenet
ATanévbenaALegjobbakASzünetek.	⇒	ktznzAabjeAevnTaaébnlgobkSüee.

Megoldási gondolatmenet:

Beolvasom a szöveget egy változóba, majd az első feladatot felhasználva „tömörítem” a szöveget, s utána kódolom a „szendvics” módszerrel, vagyis itt, amikor átmásolom a k változóba, akkor egyet az elejére, egyet a végére másolok felváltva. (Tipp: amikor felváltva akarok csinálni két utasítás(sor)t, akkor, ha nyilvántartom, hányadik lépésnél járok, akkor ez a számláló egyszer páros, egyszer páratlan lesz, így praktikus ezt vizsgálni, és ehhez kötni, mikor melyik utasítás(sor) hajtódjon végre, ebben az esetben elé vagy mögé másolom a következő karaktert.

Megoldási algoritmus:

Változók:

sz,k: szöveg;  
i: egész szám;

BE: sz;  
sz:=kódolt(sz);

k:=”;

CIKLUS i:=1-től hossz(sz)-ig  
HA (i mod 2 = 0) AKKOR  
    k:=sz[i]+k;  
KÜLÖNBEN  
    k:=k+sz[i];  
ELÁGAZÁS VÉGE  
CIKLUS VÉGE  
KI: k;

Megoldási algoritmus magyarázata:

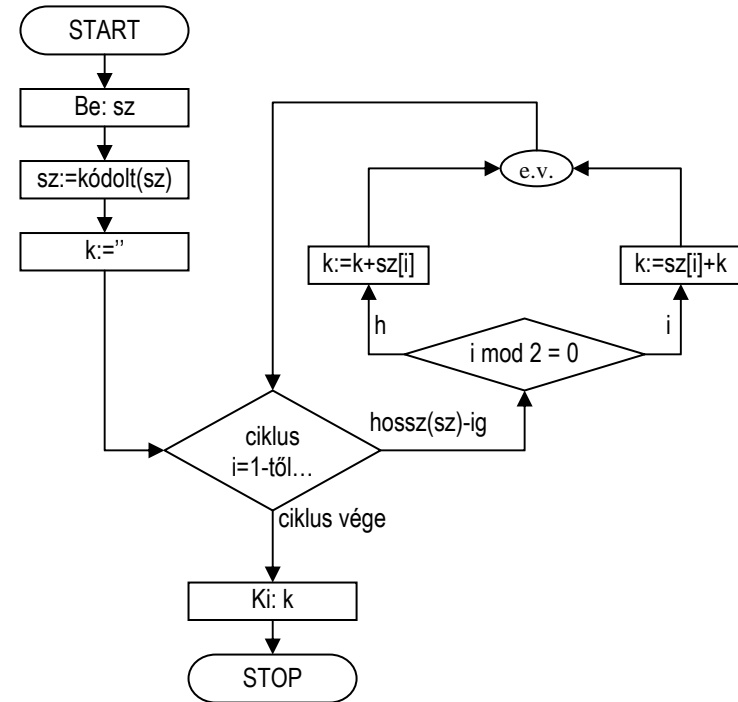
sz változóban tároljuk a beolvasott szöveget,  
k változóban tároljuk a kódolt szöveget  
i ciklusváltozóval egyesével vizsgáljuk a szöveg betűit

beolvassuk a szöveget a felhasználótól  
a beolvasott szöveget lecseréljük a „tömörítettre”  
(ide gondolatban az egész első feladat beilleszthető)  
„lenullázzuk” a k változót

az összes betűn egyhuzamban megyek végig, s  
ha i páros (vagyis 2-vel osztva 0 a maradék)  
akkor az eddigi tartalom elé írom,  
különben  
mögé írom

kiíratom a „kódolt” szöveget

Folyamatábra



3. Feladat: Állítsuk elő a(z egyben) beolvasott névhez tartozó monogramot.

Példa bemenet		Példa kimenet
Csokonai Vitéz Mihály	⇒	Cs.V.M.
Rippl-Rónai József	⇒	R.R.J.

Megoldási gondolatmenet:

Beolvasom a nevet egy változóba, majd végigmegyek a betűin: az első biztos kell, majd mindegyiket szóköz előz meg, közben persze a kettős mássalhangzókra is figyelek, vagyis ha Z után s jön, vagy C után s, vagy ... akkor a rákövetkezőt is másolnom kell.

Megoldási algoritmus:

Változók:

nev,m: szöveg;  
i: egész szám;

BE: nev;  
m:=nev[1];

CIKLUS i:=2-től hossz(nev)-ig

HA (nev[i]≠' ') ÉS  
((nev[i-1]=' ') VAGY (nev[i-1]='-'))  
AKKOR  
m:=m+nev[i];

ELÁGAZÁS VÉGE

HA ((nev[i]='C') ÉS (nev[i+1]='s')) VAGY  
((nev[i]='G') ÉS (nev[i+1]='y')) VAGY  
((nev[i]='L') ÉS (nev[i+1]='y')) VAGY  
((nev[i]='N') ÉS (nev[i+1]='y')) VAGY  
((nev[i]='S') ÉS (nev[i+1]='z')) VAGY  
((nev[i]='T') ÉS (nev[i+1]='y')) VAGY  
((nev[i]='Z') ÉS (nev[i+1]='s')) AKKOR  
m:=m+nev[i+1];

ELÁGAZÁS VÉGE

HA (nev[i]≠' ') ÉS  
((nev[i-1]=' ') VAGY (nev[i-1]='-'))  
AKKOR  
m:=m+'.';

ELÁGAZÁS VÉGE

CIKLUS VÉGE

KI: m;

Megoldási algoritmus magyarázata:

n változóban tároljuk a beolvasott nevet,  
m változóban tároljuk a monogramot  
i ciklusváltozóval egyesével vizsgáljuk a szöveg betűit

beolvassuk a szöveget a felhasználótól  
a monogram biztos úgy kezdődik mint a név, ezért...

az alábbiakat vizsgálom minden betűnél:

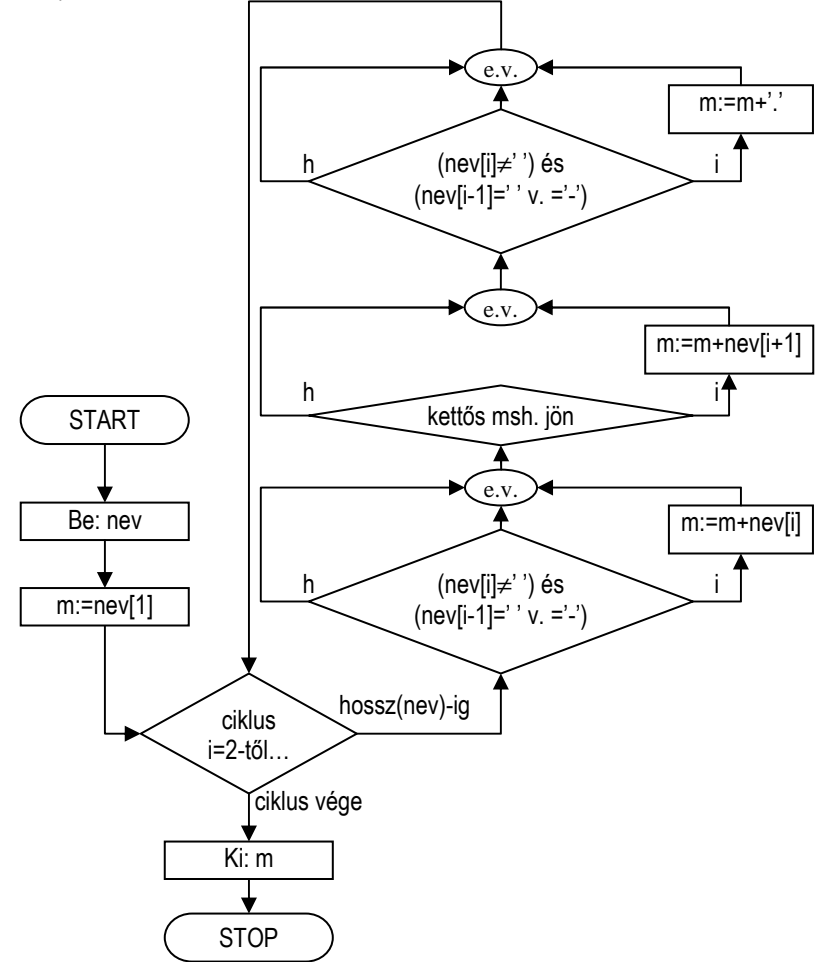
ha az nem szóköz, és  
előtte szóköz vagy kötőjel van,  
akkor  
a monogramhoz hozzáírom

ha kettős mássalhangzó van, vagyis ha  
'C' után 's' szerepel, vagy  
'G' után 'y' szerepel, vagy  
...  
...  
... akkor  
a második karakterét is átmásolom.

kell a végére egy pont is, de csak ha írtam a  
monogramhoz, azaz feltételünk u.a. mint fent  
ekkor  
egy pontot írok mögé.

kiíratom a monogramot

Folyamatábra



4. Feladat: Kérjünk be két időpontot (óra:perc) formátumban, és határozzuk meg (egy napon belül) hány perc van köztük.

Példa bemenet			Példa kimenet
3:30 és 11:15	⇒	465 (perc)	
21:00 és 6:05	⇒	895 (perc)	

Megoldási gondolatmenet:

Beolvasom a két időpontot egy-egy szöveges változóba, majd kinyerem belőlük az óra és perc értékeket. Mindkét időértéket átváltom percbé, így egy kivonással meg is van az eredmény.

Megoldási algoritmus:

Változók:

ido1,ido2: szöveg;  
o1,o2,p1,p2,kul: egész szám;

BE: ido1;  
BE: ido2;

i:=2;  
CIKLUS  
  i:=i+1;  
  AMÍG ido1[i]=':':  
  o1:=ido1[1..i-1];  
  p1:=ido1[i+1..hossz(ido1)];  
  p1:=p1+60\*o1;

i:=2;  
CIKLUS  
  i:=i+1;  
  AMÍG ido2[i]=':':  
  o2:=ido2[1..i-1];  
  p2:=ido2[i+1..hossz(ido2)];  
  p2:=p2+60\*o2;

HA p1>p2 AKKOR  
  kul:=p1-p2;  
KÜLÖNBEN  
  kil:=p2-p1;  
ELÁGAZÁS VEGE

KI: kul;

Megoldási algoritmus magyarázata:

beolvasom a két időértéket szövegesen ezekbe szedem szét az óra és perc értékeket

a 2.karaktertől kezdve...  
  ...veszem a következőt  
  az 1. időpontból...  
  ...amíg meg nincs a kettőspont  
  az óra érték a : előtti rész  
  a perc érték a : utáni rész  
  az órákat is felváltom percbé

a 2.karaktertől kezdve...  
  ...veszem a következőt  
  az 2. időpontból...  
  ...amíg meg nincs a kettőspont  
  az óra érték a : előtti rész  
  a perc érték a : utáni rész  
  az órákat is felváltom percbé

a nagyobbikból kivonom a kisebbet

kiíratom a különbséget

Folyamatábra

