

„Játékszabályok”, tudnivalók:

- A. A kódolást azzal kezdje, hogy a saját és a gyakorlatvezetőjének a nevét a forrásállományok 1. sorába (*' és '*') közé beírja. Majd kimenti **gyakorlatvezetője+saját nevéből**, és a saját **születési dátumából** „generálandó” néven, a winchesteren a saját (**GGNNhhnn** nevű) alkönyvtárába: **GGNNhhnn.PAS**, ahol GG=PG/HV/BL/ML¹, NN=2-betűs monogramja, hh=hónap 2-helyiértékű számmal ('01'..'12'), nn=nap ('01'..'31').
- B. A megoldások értékét elsősorban a **futás helyessége** határozza meg. (Szintaktikusan hibás \Rightarrow 0; elszáll \Rightarrow 0 arra a részre; \exists próba, amire jó az eredmény \Rightarrow >0, \forall próbára jó az eredmény \Rightarrow >>0.)
- C. A feladat szövegében **megadott ábrázolástól** ne térjen el!
- D. Algoritmikusan többféle megoldás is elképzelhető. Az „intelligensebb” többet ér, de egy „sablonos”, azonban teljes megoldás is ötöst ér.
- E. Az adatok beolvasásánál „első neki futásra” nem muszáj az ellenőrzést is elhelyezni, ha marad ideje, pótolja (mert valamennyi pontot azért ér).
- F. **Ne térjen el** az inputadatok (és outputadatok) tekintetében a **megadott sorrendtől, és típustól!**
- G. Felhasználhat **saját, hozott programdarabokat**, és bármilyen saját, hozott írásos anyagot. (Másét azonban semmilyen körülmények között sem!)
- H. A „**dizájn**” **nem hoz többpontot**, persze, ha a saját „preparált” kiinduló file-jában ehhez eszközök állnak rendelkezésére, bátran beillesztheti.

Feladat:

Valósítsa meg a **halmaz** típust az alábbiak figyelembe vételével:

1. Asszociált műveletei:

```

Procedure BeH(Var h:THalmazK)
{Text állományból beolvassa a halmaz elemeit; l. pl.}
Procedure KiirH(Const h:THalmazK);
{kiírja a halmaz elemeit, mint egy keresőfa elemeit; l. pl.}
Procedure Uresh(Var h:THalmazK);
Function ElemEH(Const e:TElem; Const h:THalmazK):Boolean;
Procedure HozzaadH(Var h:THalmazK; Const e:TElem);
{a halmazhoz veszi az elemet, ha még nem eleme..}
Procedure ElveszH(Var h:THalmazK; Const e:TElem);
{kiveszi a halmazból az adott elemet, ha benne volt...}
Procedure EgyesitH(Const h1,h2:THalmazK; Var h:THalmazK);
Procedure MetszH(Const h1,h2:THalmazK; Var h:THalmazK);

```

2. Reprezentációja: láncolt ábrázolású bináris keresőfa

```

Type TElem=String;
      THalmazK=TBinFa{TElem};

```

Készítse el a **halmazt megvalósító modult** (azaz vagy egy Pascal unitot, vagy egy Pascal include-állományt)! Célszerű felhasználnia –önálló modulként!– a korábban megírt **keresőfa modult**.

Modulját próbálja ki a [mellékelt „főprogrammal”](#), miután azt is a szokásos „nyitókomment tel” ellátta, s „hangolta” a saját elképzeléseihez! Mellékelnie kell ezt a **főprogramot** is, mivel a kipróbáláshoz, s így az értékeléshez szükséges.

Példa:

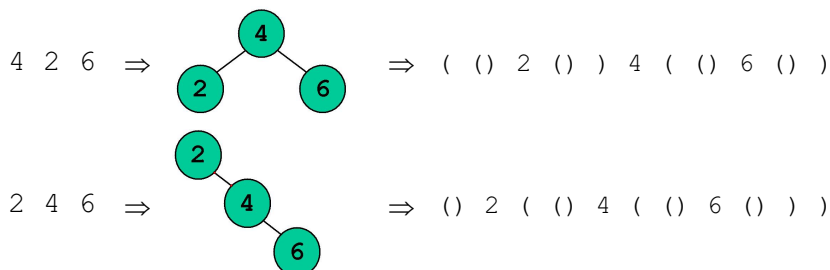
Bemeneti file (0 . INP) ² :	A kimeneti file (0 . OUT):
3	Halmaz -- keresőfára építve
1	H1:
5	((1)) 3 ((5))
–	H2:
2	(2 ((3)) 4 ((6)))
4	H1 unio H2:
3	((1 (2))) 3 ((4)) 5 ((6)))
6	H1 Metszet H2:
–	(3)

¹ PG=Pap Gáborné/HV=Heizlerné Bakonyi Viktória/BL=Bakonyi László/ML=Menyhárt László

² Az „_” az üres sort jelzi...

Vegye észre:

- az input minden adatot külön sorban vár, ami nem csak szám-típusú adat lehet (bár a próbafájl-ok csak ilyeneket tartalmaznak),
- a halmaz elemeinek felsorolását az input-file-ban egy üres sor zárja,
- mindkét próbahalmaz adatait ugyanaz a file tartalmazza,
- az output olyan lista, amelyből visszafejthető a fa szerkezet, legegyszerűbb, ha zárójel-zéssel tükrözi azt. Így:



Pontozás:

Input-alapú javítás:	Pont
A beépített adatok	5
Nem üres, diszjunkt inputsorozatok	5
Azonos, de sorrendben eltérő inputhalmazok	5
Üres az 1. inputsorozat	5
Üres mindkét inputsorozat	5
Vannak ismétlődő elemek a sorozatokban	5
Összesen:	30
Levonás jár, ha nem különíti el a modulokat!	2*(-15%)

A főprogram Ön számára lényegét jelentő részeit kiemeltük:

```

Program HalmazKeresoFaraEpiteve_Proba;
Uses Newdelay, Crt, BinFaUni;
    {$i AltRutin.inc}
Type TElem=String;
    {$i THalmazK.inc}
Var h1, h2, h: THalmazK;
Begin
    InicIO('FILE', '');
    UjLap('Halmaz -- keresőfára építve', 0);
    Writeln('H1:');
    UresH(h1);
    HozzaadH(H1, '1'); Writeln(CrLf); KiirH(h1);
    HozzaadH(H1, '2'); Writeln(CrLf); KiirH(h1);
    HozzaadH(H1, '1'); Writeln(CrLf); KiirH(h1);
    HozzaadH(H1, '3'); Writeln(CrLf); KiirH(h1);
    ElveszH(H1, '2'); Writeln(CrLf); KiirH(h1);
    ElveszH(H1, '3'); Writeln(CrLf); KiirH(h1);
    ElveszH(H1, '1'); Writeln(CrLf); KiirH(h1);
    BillreVar; Writeln('H1:');
    UresH(h1); BeH(h1);
    Writeln(CrLf); KiirH(h1); Writeln(CrLf);
    BillreVar;
    Writeln(CrLf, 'H2:');
    UresH(h2); BeH(h2);
    Writeln(CrLf); KiirH(h2); Writeln(CrLf);
    BillreVar;
    Writeln(CrLf, 'H1 Unio H2:');
    Egyesith(h1, h2, h); KiirH(h); Writeln(CrLf);
    BillreVar;
    Writeln(CrLf, 'H1 Metszet H2:');
    MetszH(h1, h2, h); KiirH(h); Writeln(CrLf);
    BillreVar;
End.

```