4. feladat: Családfa (30 pont)

Egy család történetét a családfával lehet illusztrálni, amely egy ősből (ez a fa gyökere) kiindulva mutatja a családtagok leszármazását. A családtagokat az egyszerűség kedvéért sorszámukkal azonosítjuk, az ős az 1-es sorszámú.

Írj programot (**csaladfa.pas, csaladfa.c, csaladfa.cpp**), amely megadja, hogy a családfa mikor volt a legszélesebb és mikor volt a legkeskenyebb (a fa gyökerén kívül)!

A **csaladfa.be** szöveges állomány első sorában a családtagok száma (***2≤N≤100000***) van. A 2..N. sorokban az egyes családtagok szülőjének sorszáma van, az i-edik sorban az i-edik családtagé.

A **csaladfa.ki** szöveges állomány első sorába annak generációnak a sorszámát kell írni (azok vannak ugyanabban a generációban, akik az őstől egyforma leszármazási távolságra vannak), amely a legnépesebb volt, a második sorba pedig azét, amelyik a legkisebb létszámú (de ez nem lehet az 1. generáció, azaz maga az ős). Ha több megoldás is van, akkor az őshöz legközelebbit kell kiírni!

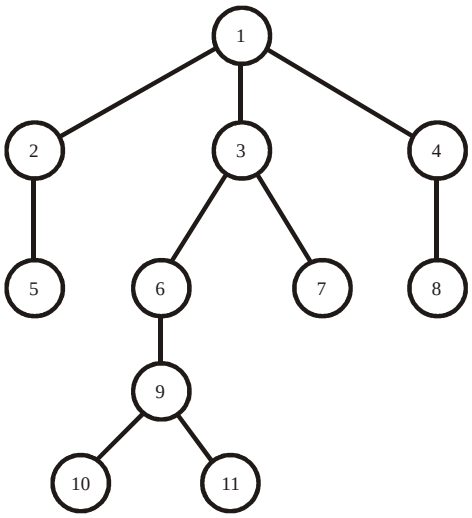
Példa:

csaladfa.be

```
11
1
1
1
2
3
3
4
6
9
9
```

csaladfa.ki

```
3
4
```



5. feladat: Szervíz (30 pont)

Mobil Szolgáltató Vállalat ***K*** város számára végez szolgáltatást. A szolgáltatást ***M*** csapat végzi. Az igényeket a beérkezés sorrendjében elégíti ki. Ismerjük ***N*** beérkezett igényt, tehát, hogy melyik városban kell elvégezni a szolgáltatást. Kezdetben az ***M*** csapat ***M*** különböző városban van. A városok egy egyenes út mentén helyezkednek el. Egy igény kiszolgálása úgy történik, hogy valamelyik csapat az aktuális helyéről elmegy az igényt kérő városba. Ennek költsége megegyezik a két város távolságával.

Készíts programot (SZERVIZ.PAS, SZERVIZ.C vagy SZERVIZ.CPP), amely kiszámítja, az összes igény optimális kiszolgálásának összköltségét, és meg is adja, hogyan kell ennek eléréséhez a kiszolgálókat mozgatni!

A SZERVIZ.BE állomány első sorában a városok ***K*** száma (***1≤K≤2000***), a szerviz-csapatok ***M*** (***1≤M≤100***) száma, valamint az igények ***N*** száma (***1≤N≤10000***) van. A városokat az ***1,…,N*** számokkal azonosítjuk. A második sorban pontosan ***K*** pozitív egész szám van (egy-egy szóközzel elválasztva), a ***K*** darab város távolsága az út kezdőpontjától számítva. A harmadik sor tartalmazza az ***M*** szerviz-csapat kezdeti pozícióját, tehát az i-edik szám annak a városnak a sorszáma, ahol kezdetben az i-edik szerviz-csapat van. A negyedik sor tartalmazza az igényeket, ***N*** város sorszámát.

A SZERVIZ.KI állomány első sorába az összes igény kielégítésének lehető legkisebb összköltségét kell írni! A második sor pontosan ***N*** egész számot tartalmazzon, az ***i***-edik szám annak a csapatnak a sorszáma legyen, amelyik az ***i***-edik igényt kielégíti! Több megoldás esetén bármelyik megadható

Példa:

SZERVIZ.BE
```
10 3 12
2 4 7 11 15 18 20 23 33 40
2 6 9
5 1 5 7 9 2 1 8 7 4 1 10
```

SZERVIZ.KI
```
36
2 1 2 2 3 1 1 2 2 2 1 3
```

6. feladat: Ültetés (40 pont)

Egy futballstadionban ***M*** ülőhelyet tartanak fenn különleges vendégek számára. Minden vendég megadhat legfeljebb négy ülőhely sorszámot, amelyekből egyet szeretne megkapni.

Készíts programot (ULTET.PAS, ULTET.C vagy ULTET.CPP), amely eldönti, hogy teljesíthető-e az összes vendég igénye, és ha igen, akkor meg is adja,!

A ULTET.BE állomány első sorában a vendégek ***N*** száma (***1≤N≤20***), és az ülőhelyek ***M*** (***1≤M≤25***) száma van. A vendégeket az ***1,…,N*** számokkal, az ülőhelyeket az ***1,…,M*** számokkal azonosítjuk. A további ***N*** sor mindegyike egy-egy vendég igényét tartalmazza, az állomány ***i+1***-edik sorában az ***i***-edik vendég által igényelt ülőhelyek sorszámai vannak felsorolva egy-egy szóközzel elválasztva és ***0***-val zárva.

A ULTET.KI állomány első sora ***N*** különböző ülőhely sorszámát tartalmazza egy-egy szóközzel elválasztva! Az ***i***-edik szám annak az ülőhelynek a sorszáma amelyiket az ***i***-edik vendég kap! Több megoldás esetén bármelyik megadható. Ha nem teljesíthető az összes vendég igénye, akkor a sor az egyetlen 0 számot tartalmazza.

Példa:

```
ULTET.BE
5 10
1 3 5 0
4 3 0
1 4 3 0
1 4 3 0
1 3 7 5 0
```

```
ULTET.KI
5 4 1 3 7
```