

1. feladat: Pártok (30 pont)

Az elmúlt 200 évben Európa 2 országában feljegyezték, hogy a választások után milyen párt (baloldali vagy jobboldali) adta a miniszterelnököt. Olyan időszakokat keresünk, amikor az összes országban ugyanolyan jellegű párt volt hatalmon. A győztes adja a miniszterelnököt a választás napjától a következő választást megelőző napig.

Feladat:

Készíts programot (PARTOK.PAS vagy PARTOK.C), amely megadja azokat az időintervallumokat a legelső választás és az aktuális idő között, amikor mindegyik országban azonos jellegű párt volt hatalmon!

Bemenet:

A PARTOK.BE állomány első sorában az aktuális dátum szerepel (év, hónap, nap, egy-egy szóközzel elválasztva). A második sorban az első országbeli választások száma van ($1 \leq N \leq 200$), a következő N sor pedig egy-egy választás eredményét tartalmazza. Utána következik a második országbeli választások száma ($1 \leq M \leq 200$), majd M sorban újra egy-egy választás eredménye. A választási eredmény első karaktere a B vagy a J betű, attól függően, hogy baloldali, vagy pedig jobboldali párt adta a miniszterelnököt, ezt követi három egész szám, egymástól és az első betűtől egy-egy szóközzel elválasztva: a választás éve, hónapja és napja.

Kimenet:

A PARTOK.KI állomány első sorába azon időintervallumok K számát kell írni, amikor mindegyik országban azonos jellegű párt adta a miniszterelnököt, a következő K sorba pedig növekvő időrendben az egyes intervallumokat. Minden intervallum 6 egész számot tartalmaz egymástól egy-egy szóközzel elválasztva: az időszak elejét (év, hónap, nap) és végét (év, hónap, nap). Ha két egymáshoz illeszkedő intervallumban ugyanaz a párt adta mindkét országban a miniszterelnököt, akkor az egy (!) intervallumnak számít!

Példa:

PARTOK.BE	PARTOK.KI
2001 5 20	2
3	1992 6 11 1994 5 11
J 1990 5 4	1996 6 30 1998 5 7
B 1994 5 12	
J 1998 5 8	
4	
J 1992 6 11	
J 1993 6 1	
B 1996 6 30	
B 1998 2 15	

2. feladat: Dominó (30 pont)

Dominóval sokféle játékot lehet játszani. Mohó Marci a következő kétszemélyes változatot kedveli. Minden dominó mindkét oldala legfeljebb 6 pöttyöt tartalmazhat, de üres is lehet. A játékban csak a 28 különböző dominó használható. A játékosok először megállapodnak, hogy az adott játékot hány dominóval fogják játszani. Jelöljük ezek számát N -el. Ez után mindkét játékos kap véletlenszerűen N (különböző) dominót, úgy, hogy legalább az egyiknek lesz dupla dominója (amelynek mindkét felén ugyanannyi pötty van). Az a játékos kezdi a játékot, akinél a legnagyobb dupla dominó van, amelyet első lépéseként le is rak. Felváltva lépnek, azonban ha a soron következő játékos nem tud lépni, akkor átadja a lépés jogát az ellenfélnek. Egy lépés abból áll, hogy a játékos egy olyan saját dominóját, amely valamelyik felével illeszkedik a már lerakott dominó-sor bal végéhez, oda rakja.

A játék akkor ér véget, ha

- Egyik játékos sem tud lépni, és ekkor egyikük sem nyer, az eredmény döntetlen
- Az soron következő játékos lerakja az utolsó dominóját, és ekkor ő nyer.

Feladat

Készíts programot (DOMINO.PAS, illetve DOMINO.C), amely meghatározza, hogy adott kezdeti játékállásra milyen kimenetele lehet a játéknak! (A feladat nem az, hogy meghatározzuk, hogy van-e és melyik játékosnak van nyerő stratégiája! Az összes lehetséges szabályos játékmenet eredménye a kérdés.)

Bemenet

A DOMINO.BE állomány első sorában a tesztesetek T ($1 \leq T \leq 10$) száma van. Minden teszteset három sort tartalmaz. Az első sorban van a játékban játékosonként kapott dominók N száma ($1 \leq N \leq 10$). A következő két sor mindegyike pontosan $2 \cdot N$ egész számot tartalmaz egy-egy szóközzel elválasztva. Az első sor az első játékos (A), a második sor a második (B) játékos dominóit írja le, egy-egy számpár egy dominót ad meg.

Kimenet

A DOMINO.KI állományba pontosan T sort kell írni, soronként a megfelelő teszteset eredményét! Az adott sor Az A betűt tartalmazza, ha bárhogy is játszanak, csak az első (A) játékos nyerhet, a B betűt, ha csak a második (B) játékos nyerhet. Az AB betűpárt (szóközők nélkül), ha mindkét játékos nyerhet, illetve a D betűt, ha egyikük sem nyerhet, bárhogy játszanak is.

Példa:

DOMINO.BE	DOMINO.KI
4	D
4	B
3 3 3 5 3 6 4 5	A
1 2 1 5 4 6 5 6	AB
4	
0 1 0 3 2 2 3 3	
1 5 3 5 5 5 5 6	
4	
3 3 3 5 3 6 4 6	
0 1 0 5 3 4 4 5	
6	

```

0 3 0 5 1 4 1 5 2 5 3 3
1 1 2 6 3 6 4 4 4 6 6 6

```

3. feladat: Lovak (40 pont)

Lótenyésztők sok generációra visszamenőleg tartják nyilván lovaik leszármazását. A lovakat sorszámukkal azonosítjuk és vagy mindkét szülőjüket ismerjük, vagy csak az egyiket, vagy pedig egyiket sem. Így ismerhetjük a lovak nagyon régi őseit is. Előfordulhat, hogy egy ló egyes ősei többféle leszármazási ágon is ősök.

Feladat:

Készíts programot (LOVAK.PAS vagy LOVAK.C), amely adott ló esetén megadja, hogy

- hány olyan őse van, amelyik több leszármazási ágon is ős;
- melyik az a ló, ami a legtöbb leszármazási úton szerepel! (Leszármazási út mindig olyan lótól indul, aminek nem ismerjük a szüleit és olyan lónál ér véget, aminek nem ismerjük az utódait.)

Bemenet:

A LOVAK.BE állomány első sorában a nyilvántartott lovak N és leszármazási kapcsolatok M száma van ($1 \leq N, M \leq 1000$). A következő M sor mindegyike 2 egész számot tartalmaz, egy szóközzel elválasztva, az első szám egy ló sorszáma, a második pedig az egyik szülőjének sorszáma. Az utolsó sorban egy ló L sorszáma van ($1 \leq L \leq N$).

Kimenet:

A LOVAK.KI állomány első sorába azon lovak számát kell írni, ahányan többszörös ősei az L lónak, a második sorba pedig azon ló sorszámat, amely a legtöbb leszármazási úton szerepel.

Példa:

LOVAK.BE	LOVAK.KI
10 11	3
10 3	3
1 2	
1 3	
2 4	
2 5	
3 5	
3 6	
5 7	
4 8	
6 8	
6 9	
1	