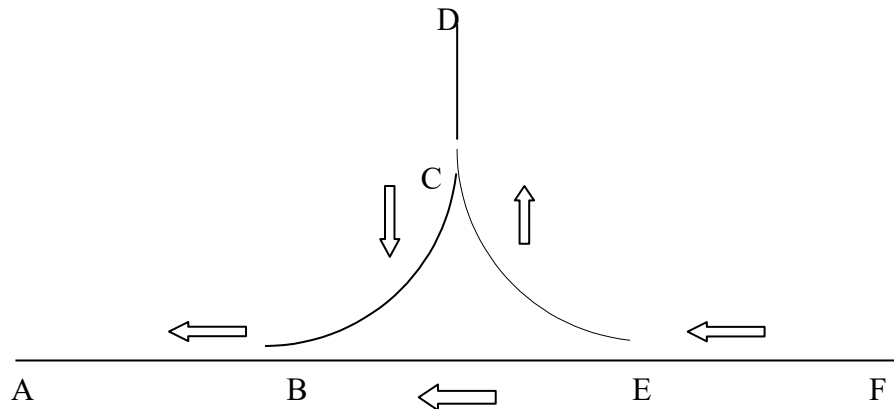


10. feladat: Vasút (50 pont)

Veremsor város vasútállomásán nagy gondot okoz a szerelvények rendezése. Az állomásról továbbítandó szerelvényeket úgy kell kialakítani, hogy amikor az megérkezik a célállomásra, a szerelvény végéről mindig lekapcsolható legyen az oda továbbított kocsisor. Minden továbbítandó szerelvény négy állomást érint, ezért a rendezés előtt minden kocsi megjelölnek az 1, 2, 3 vagy 4 számokkal. A szerelvény kocsijait rendezzük át úgy, hogy a szerelvény elején legyenek az 1-essel, aztán a 2-essel, majd a 3-assal, végül a 4-essel megjelöltek. Kezdetben a kocsik az ábrán látható E-F pályaszakaszon vannak.

A vasúti váltók működése csak a következő műveleteket teszi lehetővé. Az átrendező kocsisorból balról az első kocsit át lehet mozgatni vagy a B-E szakaszba a már ott lévő kocsik mögé, vagy a C-D szakaszba a már ott lévő kocsik elé. A B-E szakaszban lévő első kocsi átmozdítható és hozzáilleszhető az A-B szakaszon kialakítandó kocsisor végére. Hasonlóan, a C-D szakasz első (tehát az utolsónak oda érkezett) kocsija átmozdítható és hozzáilleszhető az A-B szakaszon kialakítandó kocsisor végére.



Feladat:

Írj programot (VASUT.PAS vagy VASUT.C), amely eldönti, hogy a kívánt kocsisorrend kialakítható-e az alkalmazható műveletekkel.

Bemenet:

A VASUT.BE állomány első sora a kocsik ($4 \leq N \leq 2000$) számát tartalmazza. A második sorban a rendezendő szerelvények száma áll ($1 \leq M \leq 100$). A következő M sor mindegyike egy átrendező szerelvényt ír le, N db 1 és 4 közötti egész számot tartalmaz egy-egy szóközzel elválasztva.

Kimenet:

A VASUT.KI állomány M sort tartalmaz, soronként az IGEN vagy NEM szöveget, annak megfelelően, hogy a kívánt kocsisorrend kialakítható-e a megfelelő bemeneti kocsisorrendből.

Példa:

VASUT.BE	VASUT.KI
8	IGEN
3	NEM
4 4 1 2 3 4 2 3	IGEN
4 3 2 1 4 3 2 1	

4 1 2 3 1 2 3 4

11. feladat: Számjegyzár (50 pont)

Egy cég számjegyzárakat gyárt, ami olyan eszköz, amelynek tíz gombja van az egyes számjegyek beütésére, van egy gombja (reset), amelyet megnyomva a zár alapállapotba kerül, továbbá van egy gomb (a nyitó gomb), amelyet megnyomva a zár kinyit, feltéve, hogy az egyetlen helyes jelszót billentyűztük be. Adott jelszóra egy adott chipet gyártanak, amely véges automataként működik, és ezt építik be a zárba. Ez azt jelenti, hogy ha a jelszó N számjegyből áll, akkor az automatának pontosan $N+2$ állapota van. Ha az automata egy a állapotban van és egy x jelet kap (az x számjegy gombját megnyomjuk), akkor a és x által egyértelműen meghatározott b állapotba kerül. Például, ha a jelszó az 1981, akkor a hibátlanul működő automatának az ábrán látható módon kell állapotot váltania a jelek hatására, továbbá csak az $a4$ állapotba kerülve szabad nyitnia. Az ábrán xi az összes olyan számjegyet jelöli, amely nem i , x pedig az összes számjegyet.

Feladat:

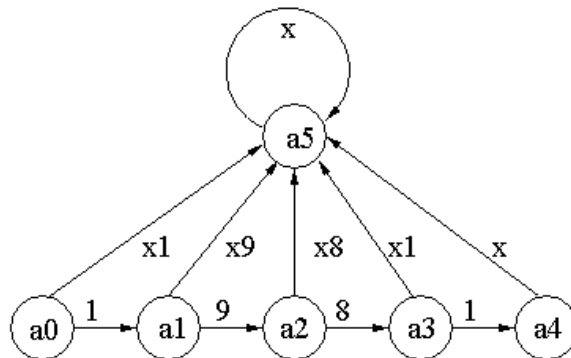
Teszteljük a kész chipet (TESZT.PAS vagy TESZT.C), hogy jól működik-e? Csak azt tudjuk ellenőrizni, hogy jelek hatására nyitó állapotba kerül-e. Tudjuk, hogy a jelszóra a zár biztosan nyit.

A teszteléshez három művelet áll rendelkezésre, amelyeket a ZAR modul valósít meg (ZAR.TPU):

ResetA: az automatát a kezdő állapotba állítja,

Be(x:Char): az adott állapotból az automata az x ($'0' \leq x \leq '9'$) jel által meghatározott állapotba megy át,

Nyito:Char: Ha az aktuális állapot nyitó, akkor a visszaadott érték 'I', egyébként 'N'.

**Bemenet:**

A TESZT.BE szöveges állomány első és egyetlen sora tartalmazza a jelszót, ami egy legfeljebb 100 decimális jelet ('0'..'9') tartalmazó karaktersorozat, amit sorvége zár. Ha a jelszó hossza N , akkor tudjuk, hogy az automatának pontosan $N+2$ állapota van.

Kimenet:

A TESZT.KI állományba egy sort kell írni. Ha a számjegyzár csak a bemenetben megadott jelszóra nyit, akkor a HIBATLAN szöveget kell írni. Ha a bemenetben megadott jelszón kívül még más kulcsra is nyit, akkor kimeneti állományba egy olyan jelsorozatot kell írni, amelyre szintén nyit a számjegyzár.