

Az interneten barangolva érdekes játékokra bukkanhatunk. Egy játékban például egy egységnégyzetekből felépített padlózaton egy $2 \times 1 \times 1$ méretű téglatestet kell úgy elgördíteni a kezdőponttól a célig, hogy eközben a téglatest minden gördítés után teljesen alátámasztva maradjon (azaz teljesen a padlózaton feködjön). A téglatest álló helyzetből indul, és álló helyzetbe is érkezik, tehát mindkét esetben valamelyik 1×1 -es lapján áll. A gördítést tekintjük a téglatest padlózattal érintkező valamelyik éle körüli 90° -os forgatásként, a (felülről nézve) jobb oldali él körüli forgatást jelöljük J-vel, a felső körülit F-fel, és így tovább.

Írjunk programot, amely meghatározza, hogy egy adott pálya megoldható-e, azaz létezik-e a téglatestet a kezdő egységnégyzetből a végsőbe vivő gördítés sorozat. Ha igen, adjunk is meg egy legkisebb számú gördítésből álló megoldást. A program a pálya leírását fájlból olvassa, az eredményt fájlba írja. A bemeneti, illetve kimeneti fájlok nevei az első, ill. második parancssori argumentumok.

A bemenet első sora két, szóközzel elválasztott, egész számot: a pálya S szélességét és M magasságát tartalmazza ($2 \leq S, M \leq 100$). Az ezt követő M sor mindegyike S számjegyet tartalmaz egymás mellé írva. Ha az adott mezőre nem terjed ki a padlózat, azaz oda nem kerülhet a téglatest, akkor ez az érték zérus. A kezdő- és végpozíciót a 2-es illetve a 3-as, a padló többi részét az 1-es számjegy jelöli.

Ha van megoldás, akkor a kimenet egyetlen sorában az J, F, B, L karakterek által kódolt gördítés sorozat, egyébként pedig a „Nem megoldható.” karaktersorozat szerepeljen.

BE	KI
7 6 0000111 0000111 2111111 0000100 0000100 0000300	JJJJFBLLLL
8 4 00000000 02111130 01100000 01100000	Nem megoldható.

Beküldendő a program forráskódja (s33.pas, s33.cpp, ...), valamint a program rövid dokumentációja (s33.txt, s33.pdf, ...), amely tartalmazza vázlatosan a megoldás leírását, és megadja, hogy a forrásállomány melyik fejlesztő környezetben fordítható.