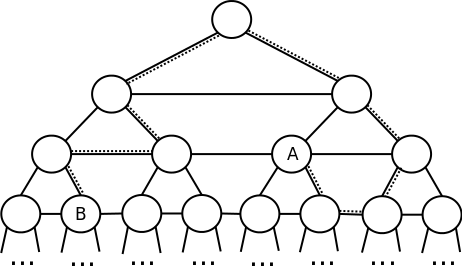
|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **Board** | **Adriatic** | **Watering** |
| **type** | batch | batch | output only |
| **time limit (per test run)** | 0,2 second | 2 seconds | - |
| **memory limit (per test run)** | 256 MB | 256 MB | - |
| **points** | 100 | 100 | 100 |
| 300 | | |

BOARD

Mirko és Slavko egy végtelen bináris fa alakú táblán játszik. Minden pontot kétirányú út köt össze az apjával, a bal és jobb fiával, valamint a szintjén belül baloldali és jobboldali szomszédjával. A gyökérpont a 0. szinten van, minden pont szintje eggyel nagyobb, mint az apjának a szintje. Minden pontnak pontosan 2 fia van.



1. ábra: A második példa ábrája

Minden gyökértől induló útvonal egy lépéssorozattal adható meg, amelynek minden lépését egy karakter azonosítja az alábbiak szerint:

* ‘1’ – lépés a bal fiúhoz,
* ‘2’ – lépés a jobb fiúhoz,
* ‘U’ – lépés az apához,
* ‘L’ – lépés a baloldali szomszédhoz,
* ‘R’ – lépés a jobboldali szomszédhoz.

Például a ‘221LU’ karaktersorozat a fenti ábrán a gyökértől az ‘A’ pontig vezető utat írja le.

FELADAT

Írj programot, amely megadja, hogy legkevesebb hány lépéssel lehet eljutni egy adott pontból egy másik pontba! A két pontot egy-egy, a gyökértől hozzájuk vezető útvonallal adják meg. Ha a két útvonal végpontja ugyanaz, akkor 0 legyen az eredmény!

INPUT

A bemenet első sorában az A ponthoz vezető útvonal van, legfeljebb 100 000 karakter.

A bemenet második sorában a B ponthoz vezető útvonal van, legfeljebb 100 000 karakter.

Mindkét útvonal szabályos.

OUTPUT

Egyetlen sorába a legkevesebb lépésszám értékét kell írni, amellyel el lehet jutni az A pontból a B pontba (vagy B-ből A-ba)!

PONTOZÁS

Legyen D az A és a B ponthoz vezető úton levő pontok szintjének maximuma!

* A tesztek 20 százalékában *D* legfeljebb 10.
* A tesztek 40 százalékában *D* legfeljebb 50.
* A tesztek 70 százalékában *D* legfeljebb 1000.

RÉSZLETES VISSZAJELZÉS

Legfeljebb 50 beküldés esetén kérhetsz részletes visszajelzést.

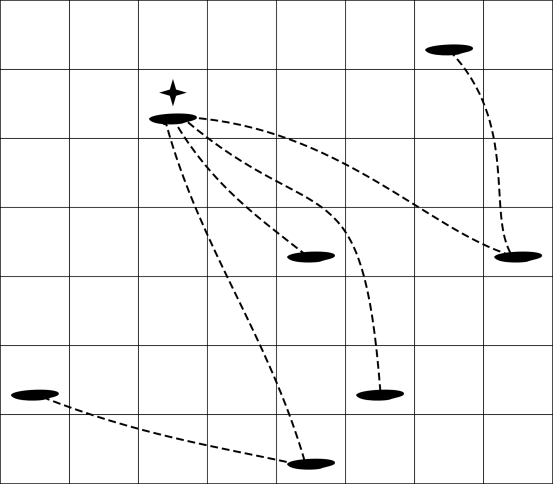
EXAMPLES

|  |  |  |
| --- | --- | --- |
| input  111RRRRRRR  222  output  0 | input  221LU  12L2  output  3 | input  11111  222222  output  10 |

ADRIATIC

Horvátország az 1000 sziget országa. A szigetek elhelyezkedését egy 2500 sorból és 2500 oszlopból álló négyzetrácson adják meg. A sorokat 1-től sorszámozzák (északról délre), az oszlopokat is 1-től (nyugatról keletre).

N sziget van, 1-től N-ig sorszámozva, mindegyik egy rácsmezőn helyezkedik el. Nincs két sziget azonos rácsmezőn. A K. szigetet a rácsmező RK sor- és CK oszlopindexével adjuk meg.



**2. ábra: az első tesztnek megfelelő térkép**

Egy lépésben hajózni csak északnyugatra vagy délkeletre lehet. Azaz az A szigetről a B szigetre egy lépésben lehet eljutni, ha *RA* < *RB* és *CA* < *CB* vagy *RA* > *RB* és *CA* > *CB*. Nem érdekes, hogy közben van-e másik sziget.

Több lépésben is el lehet jutni az A szigettől a B szigethez. A és B sziget hajózási távolságán azt a legkisebb lépésszámot értjük, amellyel el lehet jutni A-tól B-hez.

A fenti ábrán a 2. sor, 3. oszlopban levő szigettől 4 továbbiba lehet 1 lépésben hajózni, a maradék két szigethez pedig 2 lépésben.

FELADAT

Írj programot, amely minden szigetre kiszámítja a többiekhez vezető hajózási utak hosszának összegét!

A tesztek olyanok, hogy bármely szigettől el lehet hajózni valahány lépésben bármely másik szigethez.

INPUT

A bemenet első sorában a szigetek *N* száma (3 ≤ *N* ≤ 250 000) van. A következő *N* sorban a szigetek sor- és oszlopindexei vannak. Az indexek 1 és 2500 közötti egész számok.

OUTPUT

*N* sort kell kiírni! A K. sorba a K. szigettől a többiekhez vezető hajózási utak hosszának összegét kell írni!

PONTOZÁS

* A tesztek 25 százalékában *N* legfeljebb 100.
* A tesztek 50 százalékában *N* legfeljebb 1500.
* A tesztek 60 százalékában *N* legfeljebb 5000.
* A tesztek 80 százalékában *N* legfeljebb 25000.

PÉLDÁK

|  |  |
| --- | --- |
| input  7  1 7  7 5  4 5  4 8  6 6  6 1  2 3  output  16  11  12  11  12  16  8 | input  4  1 1  2 3  3 2  4 4  output  3  4  4  3 |

Watering

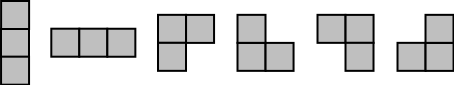
Sárának 5\**R* sorból, 5\**C* oszlopból álló négyzetrácsos földje van, függőleges és vízszintes kerítésekkel 5x5-ös részekre bontva.

Bizonyos cellákban madárijesztő van, minden 5x5-ös részben legfeljebb egy.

|  |  |  |
| --- | --- | --- |
| triomine.png | .....|.....  .....|.....  ...#.|.....  .....|.....  .....|.....  -----+-----  .....|.....  .....|.....  .....|.....  .....|.....  .....|..... | aaacc|dxxxa  bbbce|dyyya  ddd#e|dzzza  ccbae|fccbb  cbbaa|ffcdb  -----+---\_-  ssrrr|tttdd  saaax\_xxeee  yxbbb|zdaaa  yxccc|zdbbb  yxddd|zdccc |

3. ábra: Egy bemenet és az arra adható lehetséges eredmény

Meg kell oldani a föld öntözését! Öntözőfejeket kell elhelyezni úgy, hogy minden cellát öntözzünk, kivéve a madárijesztők helyét! Egy öntözőfej három cellát öntöz, az alábbi elrendezések valamelyikében:



Minden cellát csak egyetlen öntözőfej öntözhet és mindegyik öntözött cellának a földön belül kell lennie!

Egy öntözőfej által öntözött három cellának nem kell ugyanazon 5x5-ös területen belül lennie, de ha átlóg egy másik 5x5-ös részbe, akkor a közöttük levő kerítést át kell lyukasztani!

FELADAT

Minden bemenetre meg kell adni az öntözőfejek egy elrendezését, amelyek minden cellát öntöznek (kivéve a madárijesztők celláit)! A pontszámod attól függ, hogy ehhez hány helyen kellett kerítést lyukasztani.

Ez egy output only feladat. Kapsz 10 input file-t (letölthetők az értékelő rendszerből), amihez el kell készítened a megfelelő output file-okat és ezeket kell beküldeni! Az értékelő rendszer a beküldött file-t formailag ellenőrzi; ha jó formátumú, akkor kiértékeli és közli az értékelés eredményét. Tehát teljes visszajelzést kapsz minden beküldésre.

A tesztesetek olyanok, hogy mindegyikre van megoldás. Ha több megoldás van, bármelyik beküldhető.

INPUT

A bemenet első sorában az *R* és a *C* (1 ≤ *R*, *C* ≤ 100) értéke van – Sára földjének mérete.

A következő 6\**R*-1 sor soronként 6\**C*-1 karaktert tartalmaz, a föld és a kerítések leírását, cellánként egy karakterrel.

A pont karakter ‘.’ üres cellát jelöl, a ‘#’ karakter (ASCII 35) madárijesztőt jelöl. A függőleges kerítést a ‘|’ karakter (ASCII 124), a vízszintes kerítést a ‘-’ karakter (mínusz) jelöli. A ‘+’ karakter jelöli a kerítések kereszteződéseit.

Output

Az output file az öntözőfejek egy elrendezését írja le, az input file-nak megfelelő formátumban.

A kerítések lyukasztási helyét a ‘\_’ karakter jelöli. A bemenetbeli üres cellákat (ahol a bemenetben pont karakter volt) az ‘a’ – ‘z’ karakterekkel kell helyettesíteni, az alábbi szabályok szerint:

1. Egy öntözőfejjel öntözött három cellát ugyanazzal a karakterrel kell jelölni, akkor is, ha nem ugyanabban az 5x5-ös részben vannak.
2. Ugyanabban az 5x5-ös részben levő két szomszédos cellát különböző karakterrel kell jelölni, ha különböző öntözőfejekhez tartoznak.
3. Bárhova rakhatsz lyukat, de akkor különböző karakterrel kell jelölni a különböző öntözőfejekhez tartozó két szomszédos cellát, ha különböző 5x5-ös részben vannak és van köztük lyuk.
4. Egyébként különböző 5x5-ös részben levő szomszédos cellákat jelölheted ugyanazzal a karakterrel.

PONTOZÁS

Minden teszteset 10 pontos. Ha a konfiguráció nem szabályos, 0 pontot kapsz. Szabályos konfiguráció esetén:

* Ha a lyukak száma nem nagyobb, mint *R*\**C*, akkor 10 pontot kapsz.
* Egyébként 5 pontot kapsz.
* Négy tesztfile-ban minden 5x5-ös mezőn van madárijesztő.

PÉLDA

|  |
| --- |
| input  2 2  .....|.....  .....|.....  ...#.|.....  .....|.....  .....|.....  -----+-----  .....|.....  .....|.....  .....|.....  .....|.....  .....|.....  output  aaacc|dxxxa  bbbce|dyyya  ddd#e|dzzza  ccbae|fccbb  cbbaa|ffcdb  -----+---\_-  ssrrr|tttdd  saaax\_xxeee  yxbbb|zdaaa  yxccc|zdbbb  yxddd|zdccc |