# Áruszállítás üres szakaszai

Az eurázsiai vasútvonalon árut szállítanak nyugatról keletre (a másik iránnyal most nem foglalkozunk). Ismerjük a cégekről, hogy honnan hova szeretnének árut szállítani. A vonatok az első (legnyugatibb) állomásról indulnak és az utolsó (legkeletibb) állomásig haladnak. Közben minden állomáson, ahonnan valamelyik cég szállítana, felveszik az árukat, a megfelelő célállomásokon pedig lerakják. Egyszerre tetszőlegesen sok cég áruja szállítható.

Készíts programot, amely megadja azon állomásszakaszok számát, amelyeken a vonatoknak végig üresen kell menniük! Állomásszakasznak szomszédos állomáspárok egymásutánját nevezzük.

## Bemenet

A *standard bemenet* első sorában az állomások száma (1≤N≤1000000) és az áruszállítások száma (1≤M≤100000) van. A következő M sor az egyes cégek be- és kirakodási állomásának a sorszámát tartalmazza (1≤Honnani<Hovai≤N).

## Kimenet

A *standard kimenetre* azon állomásszakaszok számát kell írni, amelyeken a vonatoknak üresen kell menniük!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 10 4 8 9 1 6 1 3 3 4 | 2  Magyarázat: a 6-8. és a 9-10. állomásszakaszon nem kellett árut szállítani. |

## Korlátok

Időlimit: 0.3 mp.

Memórialimit: 32 MB

# Autókódolás

Egy autógyártó cég úgy szeretné kiosztani az autók gyártási kódjait, hogy a kódból egyértelműen meghatározható legyen az autó típusa. Azt találták ki, hogy az azonos típusú autók gyártási kódjainak kettes számrendszerbeli alakjában ugyanannyi 1-es bit legyen. Például a 3 (=112), az 5 (=1012) és a 6 (=1102) kódok mindegyikében kettő darab 1-es található. Tehát az első ilyen típusú autó a 3-as kódot kapja, a második az 5-öst, a harmadik pedig a 6-ost.

Készíts programot, amely megadja egy adott kódot közvetlenül megelőző és követő ugyanolyan típusú autók kódját!

## Bemenet

A *standard bemenet* első sorában a megadott kód van (1≤N≤1012).

## Kimenet

A *standard kimenet* első sorába a megadott kódot megelőző, a második sorába pedig a következő kódot kell írni! Ha valamelyik nem létezik, akkor a megfelelő sorba -1-et kell kiírni!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 46 | 45 51 |
| Magyarázat | |
| 46=101110 | 45=101101 51=110011 |
| Bemenet | Kimenet |
| 4294967296 | 2147483648 8589934592 |

## Korlátok

Időlimit: 0.2 mp.

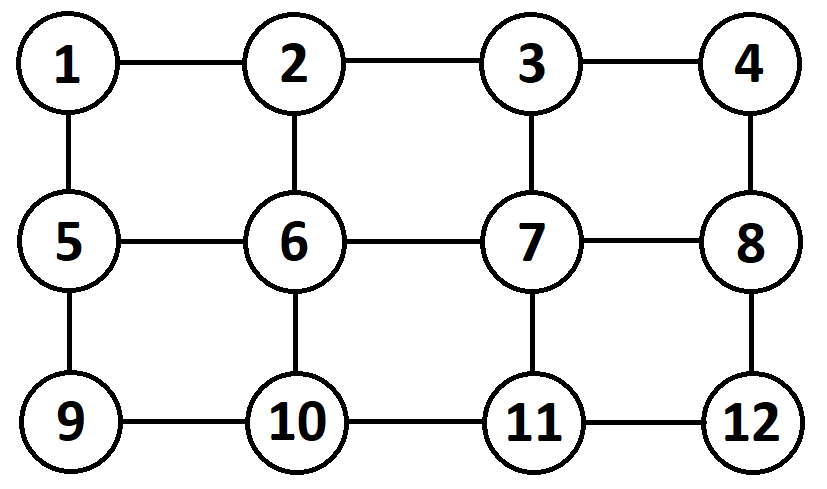
Memórialimit: 32 MB

## Pontozás

A pontszám kétharmada szerezhető olyan tesztekre, ahol N≤107.

# Rácsháló gráf

Egy rácsháló gráf négyzetrácsosan elrendezett pontokból áll, ahol minden pontból a négy szomszédjába (fel, le, jobbra, balra) vezet él, ha azok még a rácshálón belül vannak. A pontokat sorfolytonosan számozzuk:



A gráfba egyesével veszünk fel különböző új (nem rács) éleket.

Írj programot, amely minden új él felvétele után megadja a legkisebb lépésszámot, amely alatt bármely pontból bármely másik pontba el lehet jutni az élek mentén lépkedve!

## Bemenet

A standard bemenet első sorában a négyzetrács sorainak és oszlopainak száma (1≤N\*M≤200), valamint az újonnan felvett élek száma van (1≤K≤100). A következő K sorban egy-egy új él két végpontjának a sorszámai szerepelnek (1≤Xi,Yi≤N\*M).

## Kimenet

A standard kimenet i. sorába a legkisebb lépésszámot kell írni, amely alatt bármely pontból bármely másik pontba el lehet jutni az i. él felvétele után!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 3 4 4 1 6 7 10 2 12 4 9 | 5 4 4 3  Megjegyzés: az első él felvétele előtt az 1-12 és a 4-9 pontok között 5 lépést kell megtenni, bármely más pontpár között ennél kevesebbet. Az első él felvétele után a 4-9 távolság még mindig 5, a második él felvételével ez is 4-re csökken. |

## Korlátok

Időlimit: 0.5 mp.

Memórialimit: 32 MB

# Sorozat generálás

Egy sorozatot a következő szabály szerint generálunk:

Kiindulunk egy legfeljebb 2\*M jegyű pozitív egész számból, majd

* + a számot megszorozzuk A-val és hozzáadunk B-t, aminek eredményeként egy legfeljebb 4\*M jegyű pozitív egész számot kapunk;
  + ha ennél rövidebbet, akkor elölről kiegészítjük annyi 0-val, hogy 4\*M jegyű legyen.
  + A sorozat következő eleme ennek a 4\*M jegyű számnak a középső 2\*M számjegyéből alkotott szám lesz.
  + A következő lépésben ebből az új számból számolunk tovább.

A fenti lépéseket végrehajtjuk N-szer, ezáltal egy N elemű sorozatot készítünk (a kezdőszám nem része a sorozatnak).

Írj programot, amely megadja az így elkészült sorozat egymáshoz legközelebbi két egyforma elemének távolságát, valamint a sorozat elemei közül a K. legnagyobbat!

## Bemenet

A standard bemenet első sorában az M értéke (1≤M≤3), a kezdőszám (1≤X≤10M\*2-1), az A és a B értéke (1≤A,B≤10M\*2-1), a lépések száma (1≤N≤1000000), valamint a K érték (1≤K≤N) van. A bemenetbeli paraméterekkel generált sorozatban biztosan van két egyforma érték.

## Kimenet

A standard kimenet első sorába az így elkészült sorozat egymáshoz legközelebbi két egyforma elemének távolságát kell írni! A második sorba az elkészült sorozat K. legnagyobb eleme kerüljön!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 1 73 11 50 20 10 | 12 39 |
| Az így készült sorozat tagjai, pl. a két 12-es, 18-as, 24-es, ... egymástól 12 távolságra van:  85 98 12 18 24 31 39 47 56 66 77 89 2 7 12 18 24 31 39 47 | |
| Rendezve a sorozat elemeit a 10. legnagyobb az első 39-es:  2 7 12 12 18 18 24 24 31 31 39 39 47 47 56 66 77 85 89 98 | |

## Korlátok

Időlimit: 0.3 mp.

Memórialimit: 128 MB

## Pontozás

A tesztek 50%-ában M≤2 és N≤1000.

# Táblajáték

Egy játéktábla 501 sorból áll. A legfelső sorában egy mező van, majd ezután minden sorában pontosan kétszer annyi mező van, mint a fölötte levő sorban. A tábla a következő szerkezetű:



A tábla felső mezőjéről indulunk. A lehetséges lépéseket a következő számok írják le:

* 0 balra lefelé lépünk egyet,
* 1 jobbra lefelé lépünk egyet,
* 2 felfelé lépünk egyet,
* 3 balra lépünk egyet,
* 4 jobbra lépünk egyet.

Írj programot, amely beolvas egy olyan lépéssorozatot, amely elvezet a tábla valamely mezőjéhez, majd megadja, hogy ez melyik sor hányadik eleme!

## Bemenet

A standard bemenet első sorában a lépések száma van (1≤K≤500), a következő sorban pedig az egyes lépéseket leíró, összesen K darab szám (0≤Li≤4). A lépéssorozat biztosan helyes, azaz semelyik lépés eredményeként sem hagyjuk el a játéktáblát.

## Kimenet

A standard kimenet első sorába annak a sornak a számát kell írni, ahova eljutottunk (a legfelső a 0. sor)! A második sorba a soron belüli mező sorszáma kerüljön, kettes számrendszerben leírva (a bal szélső a 0. mező a sorokon belül)!

## Példa

BemenetKimenet

6 3  
0 1 4 2 1 0 110

BemenetKimenet

6 6  
0 0 0 0 0 0 0

## Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Pontozás

A tesztek 50%-ában K≤20

# Találkozás

Egy rendezvényen ismerjük az egyes vendégek érkezési és távozási időpontjait. Bárkivel találkozhatunk, amikor jelen van, beleértve az érkezésének és a távozásának az időpontjait is.

Készíts programot, amely meghatározza a legrövidebb időintervallumot, ami alatt a vendégek legalább felével találkozhatunk!

## Bemenet

A *standard bemenet* első sorában a vendégek száma van (1≤N≤100000). A következő N sor az egyes vendégek érkezési és távozási időpontjait tartalmazza (1≤Érki<Távi≤100000), érkezés szerinti sorrendben.

## Kimenet

A *standard kimenet* első sorába a legrövidebb olyan időintervallum időpontjai számát kell írni, ami alatt a vendégek legalább felével találkozhatunk! A második sorba ennek az intervallumnak a kezdő- és végidőpontját kell kiírni (több megoldás esetén a legkorábbit)!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 6 1 5 2 3 5 9 7 8 10 15 10 18 | 2 9 10  Magyarázat: a 3. vendéggel még a 9. időpontban találkozhatunk, az utolsó kettővel pedig már a 10.-ben. |

## Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB