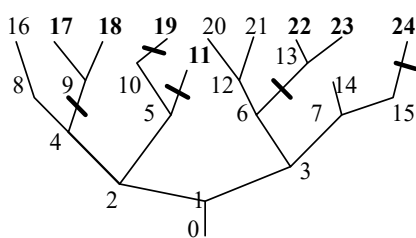


10. feladat: Favágás (19 pont)

5
0 0 1 -1
1 0 2 3
2 0 4 5
3 0 6 7
4 0 8 9
5 0 10 11
6 0 12 13
7 0 14 15
8 0 16 -1
9 0 17 18
10 0 -1 19
11 1 -1 -1
12 0 20 21
13 0 22 23
14 0 -1 -1
15 0 -1 24
16 0 -1 -1
17 1 -1 -1
18 1 -1 -1
19 1 -1 -1
20 0 -1 -1
21 0 -1 -1
22 1 -1 -1
23 1 -1 -1
24 1 -1 -1

Egy gyümölcsös bináris birsalmafaának leveleit betegség támadta meg. Szerencsére a fertőzés lassan terjedt. A gyümölcsös gazdája időben észlelte a dolgot. A betegség még nem jutott el más ágakra. A beteg ágak levágásával a fák még megmenthetők.

Minden fára ismerjük a vágások számát. (Természetesen ez nem lehet nagyobb a beteg levelek számánál, de legalább egy.)



Írj a gazdának egy olyan programot, ami megadja minden fához, hogy mely ág(ak)at kell elmetszeni ahhoz, hogy az összes beteg levelet levágjuk, de a megmaradt fának a lehető legtöbb ága (éle) legyen.

A fa legalább egy ággal rendelkezik. A gyökérelemből csak egy (bal oldali) ág indul ki.

Programod neve: FAK.PAS vagy FAK.CPP.

Bemenő adatok:

- A FAK.BE állomány első sorában a vágások száma van (maximum 20),
- majd a csomópontok következnek (nem feltétlenül rendezetten).

Egy csomópontot négy számmal adunk meg:

- A első a csomópont azonosítási száma (0-tól 499-ig vehet fel értékeket). A 0 jelenti a gyökérelemet.
- A második 0 vagy 1 lehet. Az 1 jelenti azt, hogy a levél fertőzött.

- A harmadik a bal oldali leszármazott azonosítója (vagy -1 , ha nincs neki)
- A negyedik a jobb oldali leszármazott azonosítója (vagy -1 , ha nincs neki)

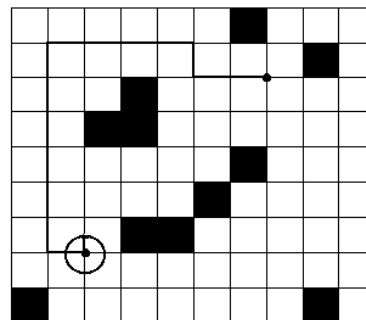
Kimenő adatok:

9	19	11	13	24
---	----	----	----	----

A FAK.KI állományba egy sort kell írni, ami a vágásokat tartalmazza. Egy vágást a csomópont (ez alatti ágat kell elmetszeni) azonosítójával kell megadni. A számokat egy szóköz választja el egymástól.

11. feladat: Robot (16 pont)

Egy raktárban robotot üzemeltetnek anyagmozgatásra. A robot csak kijelölt pályákon mozoghat. A pályák rácsos szerkezetet alkotnak, a szomszédos pályák távolsága 1 méter. A raktár téglalap alakú, mérete $N \times M$ méter. A szélső robotpályák a raktár falától 1 méter távolságra vannak. A robot alakja kör, aminek átmérője 1.6 méter. A robot középpontjával van a pályán és mindig csak egy rácspontban állhat meg. A raktárban a robot mozgását akadályok gátolják, ezek mindegyike egy négyzetet foglal el. A robot a következő utasításokat tudja végrehajtani, mindegyiket 1 másodperc alatt.



- Előre: a pillanatnyi irányításnak megfelelően 1 métert halad előre,
- Fordul balra: irányítását 90° -al balra fordítja,
- Fordul jobbra: irányítását 90° -al jobbra fordítja.

Készíts programot, amely kiszámítja, hogy egy adott pontból egy adott másik pontba a robot hány másodperc alatt tud eljutni a leggyorsabban.

Programod neve: ROBOT.PAS vagy ROBOT.CPP.

Bemenő adatok

A ROBOT.BE állomány első sora az N és M értékeket tartalmazza ($N, M \leq 50$). A második sorban négy pozitív egész szám van, x_1 y_1 x_2 y_2 , ahol (x_1, y_1) az indulási pont, (x_2, y_2) pedig a cél pont koordinátái. A további N sor mindegyike M számot tartalmaz, 1-et, ha az adott négyzet akadály, 0 egyébként. Egy akadályt az általa elfoglalt négyzet bal felső sarkának koordinátáival adunk meg. Sor-oszlop koordinátarendszert használunk, azaz a bal felső négyzet koordinátái $(0,0)$, a jobb alsóé pedig $(N-1, M-1)$. A robot kezdetben a felső sor felé néz, célba érkezve irányítása tetszőleges lehet.

Kimenő adatok

A ROBOT.KI állomány egyetlen számot tartalmazzon, azt, hogy a robot leggyorsabban hány másodperc alatt tud eljutni az indulási pontból a cél pontba. A -1 értéket kell kiírni, ha az akadályok miatt nem tud eljutni a cél pontba a robot.

Példa: ROBOT.BE

```

9 10
7 2 2 7
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0 0
0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 1 0

```

ROBOT.KI

```
19
```

12. feladat: Lámpák (16 pont)

Egy terem világítását $N \cdot M$ lámpa biztosítja, amelyek a mennyezetten N sorban és M oszlopban vannak elhelyezve. Minden sorhoz és oszlophoz tartozik egy-egy nyomógomb. ha lenyomjuk, az adott sor (vagy oszlop) összes lámpája ellenkező állapotba vált (kialszik, ha égett, illetve fordítva). Kezdetben egyik lámpa sem ég. Készíts programot, amely egy előre megadott világítási konfigurációhoz (melyik lámpa világít és melyik nem) megadja, hogy minimálisan hány kapcsolással lehet előállítani a konfigurációt. Az eredmény -1 , ha nincs megoldás.

Programod neve: LAMPAK.PAS vagy LAMPAK.CPP.

Bemenő adatok:

A LAMPAK.BE szöveges állomány első sorában a sorok és az oszlopok száma van, egy szóközzel elválasztva ($1 \leq N, M \leq 100$). A következő N sor mindegyikében M db számmal írjuk le –egy-egy szóközzel elválasztva– az adott sor világítási konfigurációját: a szám 1 , ha a lámpának égni kell, 0 , ha nem.

Kimenő adatok:

A LAMPAK.KI szöveges állományba a világítási konfiguráció előállításához szükséges minimális kapcsolások számát kell írni.

Példa: LAMPAK.BE

```
3 5
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
```

LAMPAK.KI

```
4
```