

Kincsek

Az Adrián egy új sziget keletkezett, amelyen kincsek találhatók. A sziget $N \times N$ -es négyzetrácsban elhelyezkedő cellákból áll. A sorokat fentről lefelé, az oszlopokat balról jobbra sorszámozzuk. Bizonyos cellák kincset tartalmaznak. Szeretnénk tudni, hogy melyek ezek a cellák. Kérdéseket tehetsz fel tetszőleges téglalapra, amelyre válaszul megkapod, hogy a téglalap hány cellája tartalmaz kincset.

Egy kérdés költsége $1 + N \times N - S$, ha a téglalapban S cella van.

Írj programot, amely kideríti, hogy mely cellák tartalmaznak kincset! A cél, hogy minél kevesebb költséggel határozd meg, de nem szükséges a minimálisat elérned!

Kommunikáció

A programodnak egy könyvtári modult kell használnia, amely kérdéseket és a válaszadást valósítja meg.

- Először a méretet kell beolvasnod a `get_n()` ; függvénnyel ($2 \leq N \leq 100$)!
- A kérdésekhez az R_1 , C_1 , R_2 és C_2 számokat kell paraméterként átadnod ($1 \leq R_1 \leq R_2 \leq N$, $1 \leq C_1 \leq C_2 \leq N$) a `count` függvénynek! Hibás kérdés esetén a programod 0 ponttal befejeződik.
- Válaszként azon (R, C) sor- és oszlopindexű cellák számát kapod, amelyekre teljesül, hogy a kérdésben szereplő téglalapban vannak, azaz $R_1 \leq R \leq R_2$ és $C_1 \leq C \leq C_2$ és az (R, C) kincset tartalmaz.
- Amikor a programod befejezi a kérdéseket, akkor az `END()` ; függvényt kell meghívnod! Ez kiír egy sort a `standard kimenet`-re, ami a kérdések összköltségét tartalmazza.
- Majd a megoldásodat a `standard kimenet`-re kell kiíratnod. N sor mindegyikébe N karaktert kell kiírni. Az i -edik sor j -edik karaktere 1, ha (i, j) cellában van kincs, egyébként pedig 0.

Felteheted, hogy minden kérdésre helyes választ kapsz, ami nem függ a feltett kérdések sorrendjétől, azaz előre rögzített, hogy hol van kincs.

Kimenet

A `standard kimenet` első sorába azon pontok számát kell írni, amelyek úgy köthetők össze egy egyenes vezetékekkel az origóval, hogy ez a pont az egyetlen közös pont az áramkör vezetékeivel! A második sor ezeket a pontokat sorolja fel, növekvő sorrendben!

Példa

A következő példában bal oldalon vannak a kiírt kérdések és a megoldás, a jobb oldalon pedig N értéke és a kapott válaszok.

függvény hívások	viSSzatérési értékek
<code>get_n();</code>	2
<code>get_count(1, 1, 1, 1);</code>	0
<code>get_count(1, 2, 1, 2);</code>	1
<code>get_count(2, 1, 2, 2);</code>	2
<code>END();</code>	

Az `END()`; függvény kiírja a standard *kimenet*-re a kérdések költségét,

```
11 <-libPrintsCost
```

ami után még ki kell íratnod a következő sorokat:

```
01
```

```
11
```

Korlátok

Időlimit: 1 mp.

Memórialimit: 256 MB

Pontozás

Minden teszteset 10 pontos. Ha a programod nem helyes eredményt ad, akkor 0 pontot kapsz. Különben a pontszám az alábbiak szerint függ a K költségtől:

- Ha $K \leq 7/16 * N^4 + N^2$, akkor a pontszámod 10 pont,
- különben ha $K \leq 7/16 * N^4 + 2 * N^3$, akkor 8 pont,
- különben ha $K \leq 3/4 * N^4$, akkor 4 pont,
- különben ha $K \leq N^4$, akkor 1 pont,
- különben 0 pont.

Az elérhető pontok 40%-ában N legfeljebb 20.

A megoldás helyességét a kérdésektől függetlenül ellenőrzik (akár kérdések nélkül is adhatsz megoldást).