

Az alábbi feladatban egy olyan faluvégi utcát vizsgálunk, amelynek csak az egyik oldalán laknak. A helyi szokásoknak megfelelően minden portát festett fakerítés határol, a ház pedig a kerítéstől kicsit bentebb található. Egy-egy porta kerítése egyszínű.

Az utca portáit leíró adatokat az `utca.txt` állomány tartalmazza. Az állomány sorai az utca portáit valós elhelyezkedésük sorrendjében (az alsó végtől a felső végig) adják meg. A sorok két számot és egy, legfeljebb 20 karakteres szót tartalmaznak. Az első szám a porta utcafront oldali hosszát adja meg méterben (másik kiterjedésükben egyeznek), a másik pedig a kapunak a porta sarkától mért távolságát. (A most pontszerűnek tekintett kapu soha nem kerül telekhatárra.) A szó a kerítés színét adja meg. Az utca hossza legfeljebb 10 km.

Például:

```
15 3 sarga
18 5 kek
11 4 sarga
...
```

A 2. sor szerint az utca alsó végtől számított második portájának utcahossza 18 méter, a sarkától 5 méterre, az utca elejétől 20 méterre van.

Készítsünk programot, amely az alábbi kérdésekre válaszol. A program forráskódját `porta` néven mentjük.

A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írjuk a képernyőre a feladat sorszámát. Amennyiben a felhasználótól kérünk be adatot, jelenítsük meg a képernyőn, hogy milyen értéket várunk (például 4. feladat: Kérem a lámpaoszlopok távolságát!).

1. Olvassuk be az `utca.txt` állományban talált adatokat, s annak felhasználásával oldjuk meg a következő feladatokat.
2. Adjuk meg, hány olyan telek van az utcában, amelynek kerítése piros színű.
3. Határozzuk meg, hogy a legnagyobb telek területe hányszorosa a legkisebbnek. Az eredményt két tizedes pontossággal jelenítsük meg.

A faluban most készül a közvilágítás. Az utca kezdetére, a sarokra kerül egy villanyoszlop, majd v méterenként egy-egy újabb. Minden lámpa a kerítésnek ugyanolyan hosszú szakaszát képes megvilágítani, az előtte lévő l métert és az utána lévő l métert. v és l egész számok, valamint $v \geq 2l$, azaz a fénykörök legfeljebb érintik egymást.

4. Kérjük be a felhasználótól v és l értékét. A további feladatok megoldásánál ezeket az értékeket használjuk.
5. Állítsuk elő az utca villanyoszlopokkal kiegészített képét a `kep.txt` állományban. Az első sorban a kerítések jelenjenek meg, a másodikban a villanyoszlopok. A telekhatárokat a szűrő karakter (`|`) jelezze, a kaput szóköz. A kerítést színének első karaktere jelölje. A villanyoszlopokat az `0` betű adja meg. A feladat elején szereplő példának az alábbi képrészlet felel meg 6 méteres oszloptávolság esetén (csak a két szürke háttérű sort kell a kimenetnek tartalmaznia):

```
012345678901234567890123456789012345678901234
|ss sssssssssss|kkkk kkkkkkkkkkk|sss ssssss|
0      0      0      0      0      0      0      0
012345678901234567890123456789012345678901234
```

6. A világítás sokba kerül, ezért az utca lakói elhatározzák, hogy amíg senki nincs az utcán, csak annyi lámpát tartanak égve, amennyi elegendő az összes kapu megvilágításához. A bekapcsolt lámpák sorszámának meghatározásához a következő algoritmust is használhatjuk.

```
Ciklus amíg van meg nem világított kapu
  Keresd meg az első ilyen kaput!
  Határozd meg, melyik az utolsó lámpa, ami megvilágítja!
Ciklus vége
```

A bekapcsolt lámpák sorszámát egymástól szóközzel elválasztva jelenítsük meg a képernyőn.

7. A lakók ráébredtek, hogy ez még mindig túl sokba kerül, ezért arra jutottak, hogy egy lámpa csak addig világítson, amíg valaki ebben az utcában tartózkodik, és a fénykörében van. Határozzuk meg, hogy mennyi az egyes lámpák világítási idejének összege, ha valaki 100 méter/perc sebességgel sétál végig az utcán közvetlenül a kerítés mentén. A világítási időt percben jelenítsük meg a képernyőn.

Beküldendő egy tömörített `i253.zip` állományban a program forráskódja (`i253.pas`, `i253.cpp`, ...), valamint a program rövid dokumentációja (`i253.txt`, `i253.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztő környezetben fordítható.