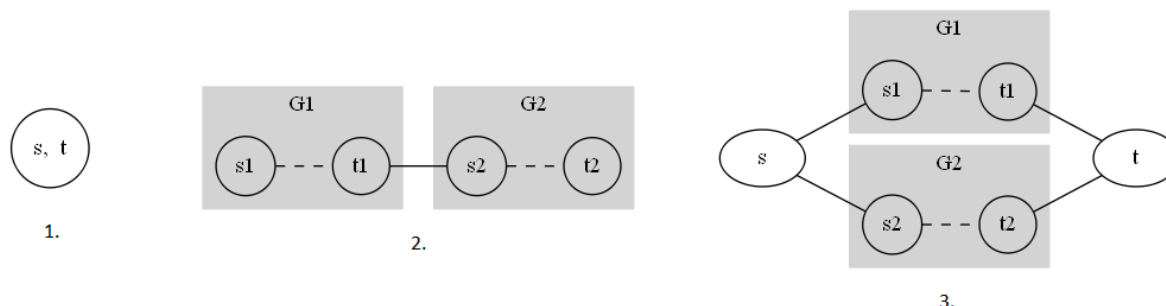


## Parkoló

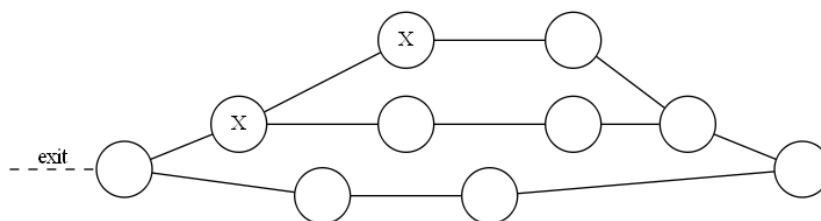
Olyan parkoló-rendszert építettek az Adria partján, amely soros-párhuzamos gráf konstrukcióval írható le. A parkoló-rendszer parkolóhelyekből és azokat összekötő kétirányú útszakaszokból áll. Bármely kettőt legfeljebb egy útszakasz köt össze. Bármely parkolóhelyen legfeljebb egy autó állhat. Foglalt helyeken nem lehet áthaladni.



1. ábra: A szabályoknak megfelelő ábrák

Minden soros-párhuzamos parkoló tartalmaz két kitüntetett helyet, ezek neve kijárat ( $s$  jelöli az ábrákon) és terminál ( $t$  jelöli az ábrákon). A soros-párhuzamos parkoló az alábbi 3 szabály ismételt rekurzív alkalmazásával kapható. A szabály tartalmazza a leírás formáját is:

1. Az egyetlen helyet tartalmazó parkoló szabályos. Ez a kijárat és egyúttal a terminál is. Az ilyen parkolóhelyet a leírásban a kis 'o' betű jelöli, ha a hely üres, illetve a kis 'x' betű, ha a hely foglalt (azaz autó áll ott).
2. Ha  $G1$  és  $G2$  szabályos parkoló, akkor a  $G$ -vel jelölt soros kompozíciójuk is szabályos, amelyet úgy kapunk, hogy a kijárata a  $G1$  kijárata, terminálja a  $G2$  terminálja, és a  $G1$  terminálja és a  $G2$  kijárata között kétirányú útszakaszt vezetünk be. Ha  $E1$  és  $E2$  a  $G1$  és  $G2$  leírása, akkor a  $G$  leírása az 'SE1E2#', azaz a leírás nagy 'S' betűvel kezdődik, ezután jön a  $G1$  és  $G2$  leírása, a végét pedig a '#' karakter jelzi.
3. Ha  $G1$  és  $G2$  szabályos parkoló, akkor a  $G$ -vel jelölt párhuzamos kompozíciójuk is szabályos, amelyet úgy kapunk, hogy bevezetjük az  $s$  és  $t$  új parkolóhelyeket, amelyek közül az  $s$  lesz a  $G$  kijárata,  $t$  pedig a terminálja. Bevezetünk 4 új kétirányú útszakaszt:  $s-s1$ ,  $s-s2$ ,  $t1-t$ ,  $t2-t$ , ahol  $s1$  a  $G1$ -nek,  $s2$  pedig a  $G2$ -nek a kijárata,  $t1$  a  $G1$ -nek,  $t2$  pedig a  $G2$ -nek a terminálja. Ha  $E1$  és  $E2$  a  $G1$  és  $G2$  leírása, akkor a  $G$  leírása a 'PEs|E1E2|Et#', ahol az  $Es$  az új kijárat leírása,  $Et$  az új terminál leírása (mindegyik a foglaltságnak megfelelően vagy kis 'o' betű vagy kis 'x' betű). A leírás nagy 'P' betűvel kezdődik, ezt követi az új kijárat, majd a '|' függőleges vonal. Ezután jön a két ág leírása, amit újabb '|' követ. Végül az új terminál leírása és a '#' karakter következik.



1. ábra: Az első tesztesetnek megfelelő parkoló gráfja

A fenti ábra leírása: 'Po|Px|Sxo#Soo#|o#Soo#|o#'. Megjegyzendő, hogy pontosan annyi parkolóhely van, ahány kisbetű szerepel a leírásban. Az egyetlen kimenetet, ahol az autók a parkolót elhagyhatják, az ábrán az exit felirat jelöli.

Egy parkolóhelyen álló autó nem blokkolt, ha van olyan útvonal, amely a parkolóhelyétől a kijáráshoz vezet olyan helyeken keresztül, amelyek nem foglaltak. A fenti ábrán egyik autó helye sem blokkolt. Ha lenne egy autó a parkoló terminálján (a legjobboldalibb helyen), akkor az egy autót blokkolna (a felsőt). Ha a kijáráshoz tennénk autót, akkor az minden más autót blokkolna.

Úgy szeretnénk autókat elhelyezni, hogy egyik sem legyen blokkolt. Lehetnek már autók a parkolóban, de ezekről tudjuk, hogy egyik sem blokkolt.

Írj programot, amely kiszámítja az autók maximális számát, amelyek elhelyezhetők úgy, hogy egyik se legyen blokkolt! Ebbe a számba beleértjük a már ott levő autók számát is. Meg kell adni továbbá a maximális számú autó egy lehetséges elhelyezését, a már ott levőket nem lehet elmozdítani!

## Bemenet

A *standard bemenet* első sora egy soros-párhuzamos parkolóleírást tartalmaz, amely legalább 1 és legfeljebb 100 000 karaktert tartalmaz. A leírásban csak az alábbi karakterek lehetnek: nagy P és S, kis o és x, # (ASCII 35) és | (ASCII 124) karakter. Ebben biztos nincs blokkolt autó.

## Kimenet

A *standard kimenet* első sorába az autók maximális számát kell írni, amelyek elhelyezhetők úgy, hogy egyik se legyen blokkolt! A második sorba a maximális számú autó egy lehetséges elhelyezését kell írni, amelyben pontosan M darab kis x kell legyen, továbbá az új x-et csak valamely bemenetbeli o helyére tehetesz! Több megoldás esetén bármelyik megadható.

## Példa

Bemenet

```
Po | Px | Sxo#Soo# | o#Soo# | o#
```

Kimenet

```
3
```

```
Po | Px | Sxo#Sox# | o#Soo# | o#
```

Bemenet

```
Po | SPo | oo | o#Px | oo | o##Po | Sxo#Po | ox | o# | o# | o#
```

Kimenet

```
7
```

```
Po | SPo | xx | o#Px | ox | o##Po | Sxx#Po | ox | o# | o# | o#
```

## Tesztelés

A `splot_tool` (amit letölthetsz az értékelő rendszerből) SVG képet készít egy szabályos leírásból, amit böngészővel nézhetsz meg. Használata:

```
$ ./splot_tool splot.dummy.out.1
splot.dummy.out.1 parsed (10 parking spaces).
splot.dummy.out.1.svg created.
$ chromium splot.dummy.out.1.svg
```

A program ellenőrzi, hogy szabályos-e a megadott leírás, de más ellenőrzést nem csinál. Legfeljebb 200 karakteres lehet a leírás.

### **Korlátok**

Időlimit: 2 mp.

Memórialimit: 256 MB

### **Pontozás**

30 pont szerezhető olyan bemenetekre, ahol a parkolóhelyek száma legfeljebb 20.

További 40 pont szerezhető olyan bemenetekre, amelyekben nincs foglalt hely, azaz a bemenetben nincs x betű.