

Hoeken

Probleem

Schrijf een programma dat een ASCII figuur produceert. Het programma heeft een parameter n die de grootte van de figuur bepaalt. Gegeven zijn de figuren voor enkele waarden van n . Je moet zelf afleiden hoe de figuur schaalt met n .

Invoer

De invoer is een geheel getal $n \in [1, 100]$, dus inclusief 1 en 100.

Uitvoer

De gevraagde ASCII figuur moet afgedrukt worden op de standaard uitvoer. De uitvoer mag een of meer lege regels bevatten voor en na de regels van de eigenlijke figuur. Op het einde van elke regel mogen een willekeurig aantal spaties staan. Het aantal spaties aan het begin van de regel is wel van belang: het meest linkse * of | karakter staat tegen de linker marge en wordt niet voorafgegaan door een spatie.

Voorbeelden

Invoer:

1

Uitvoer:

```
* *  
*  
* *
```

Invoer:

2

Uitvoer:

```
** **
*  *
  **
  **
*  *
** **
```

Invoer:

3

Uitvoer:

```
*-*  *-*
|/   \|
*    *
  *-*
  | |
  *-*
*    *
|\   /|
*-*  *-*
```

Invoer:

4

Uitvoer:

```
*--*  *--*
| /   \|
| /   \|
*    *
  *--*
  | |
  | |
  *--*
*    *
|\   /|
| \  /|
*--*  *--*
```

Om te zorgen dat je zeker weet waar en hoeveel spaties er moeten staan, staat de uitvoer voor invoer 4 hierna nog eens met elke spatie vervangen door een b:

```
*--*bbbb*--*
|b/bbbbbbb|
|/bbbbbbbb|
*bbbbbbbbb*
bbbb*--*
bbbb|bb|
bbbb|bb|
bbbb*--*
*bbbbbbbbb*
|\bbbbbb|
|b\bbbbbb|
*--*bbbb*--*
```

Schildpad

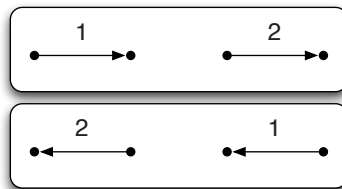
Het ministerie van wegenwerken heeft het schilderen van wegmarkeringen geautomatiseerd met Turbot, de robotschildpad ontwikkeld door prof. Harry Askill. Turbot voert nauwgezet een programma van instructies uit om wegmarkeringen te schilderen.

Na uitvoerige experimenten hebben de ingenieurs van het ministerie vastgesteld dat een enkele laag verf niet volstaat. Als Turbot 's morgens net de centrale verlaten heeft is de intensiteit van de verf veel hoger dan 's avonds vlak voor hij terugkeert. Daarom stellen de ingenieurs voor dat Turbot elke tweede dag een bijkomende laag verf aanbrengt op die van de vorige dag. De verf moet dan wel in omgekeerde volgorde aangebracht worden om overal een gelijke intensiteit te bekomen.

Moeten de ingenieurs nu twee verschillende programma's schrijven, voor de eerste *en* de tweede dag?

Probleem

Schrijf een programma dat, gegeven een programma voor de eerste dag, een equivalent programma voor de tweede dag maakt dat de markering in omgekeerde volgorde schildert. Twee programma's zijn equivalent als ze dezelfde markering tekenen. Een markering wordt in omgekeerde volgorde getekend als de laatste lijn eerst getekend wordt, de voorlaatste lijn tweede enz. Bovendien moet elke lijn van eindpunt naar beginpunt getekend worden. Hieronder zie je een voorbeeldmarkering bestaande uit twee lijnstukken, en zijn omgekeerde.



Invoer

De eerste lijn van de invoer is het aantal lijnen n met instructies voor Turbot dat volgt. Hierbij is $n \in [1, 100]$. De volgende n lijnen bevatten elk een instructie voor Turbot. De mogelijke instructies zijn:

-
- b borstel op/nee: zet de verfborstel neer op de weg als hij op was, of trek hem omhoog als hij neer was.
 - s zet een stap van 1m vooruit; als de verfborstel neer staat wordt hierbij een lijn getekend.
 - d draai ter plaatse 90° naar rechts.
 - h beëindig het programma.

Turbot vertrekt bij aanvang van het programma steeds op dezelfde locatie, de *central*. Initieel is de borstel van Turbot omhoog. Een programma markeert dezelfde lijn nooit meer dan eens. De h instructie komt enkel, en altijd, als laatste voor.

Uitvoer

De uitvoer bestaat uit m lijnen met een equivalent omgekeerd programma voor Turbot.

Voorbeelden

Invoer:

3
b
s
h

Uitvoer:

s
d
d
b
s
h

Invoer:

7
b
s
b
s
b
s
h

Uitvoer:

s
s
s
d
d
b
s
b
s
b
s
h

Invoer:

7
b
s
d
s
d
s
h

Uitvoer:

d
s
d
d
d
b
s
d
d
d
s
d
d
s
h

De slimme postbode

Pietje Rolog, de postbode van Heverlee, bedielt de post in de Celestijnenlaan. Hij heeft het niet zo begrepen op die nieuwe efficiënte postrondes. Het liefst van al zou hij de hele dag bezig zijn met het bedelen van post in dezelfde straat. Hij heeft al een idee hoe hij dat gaat aanpakken. In plaats van de brieven te bezorgen in volgorde van oplopende huisnummers, gaat hij ze in een andere (hopelijk langere) volgorde bedelen. Jij gaat hem erbij helpen om de langst mogelijke route te vinden, of ten minste een langste route als er meerdere zijn. Er staan n huizen in de straat met huisnummers van 1 tot n . Je moet de huisnummers in die volgorde teruggeven waarvoor de totale afstand tussen de opeenvolgende huizen maximaal is. De afstand tussen twee huizen is het verschil in hun huisnummers.

Bijvoorbeeld, bij 3 huizen, zijn de huisnummers 1, 2 en 3. Als Pietje Rolog ze in volgorde bezoekt, legt hij een afstand af van $(2 - 1) + (3 - 2) = 2$ eenheden. De volgorde 1, 3, 2 is langer omdat hij dan $(3 - 1) + (3 - 2) = 3$ eenheden moet afleggen. Omdat er geen langere volgorde is, is dit dus een gezochte volgorde.

Merk op dat de afstand van het begin van de straat tot het eerst bezochte huis en van het laatst bezochte huis tot het einde van de straat niet meetellen. Pietje wordt namelijk 's morgens aan het eerste huis afgezet met het post-minibusje, en 's avonds aan het laatste huis opgepikt.

Probleem

Gegeven n , zoek een permutatie $(x_1 \cdots x_n)$ van $(1 \cdots n)$ waarvoor

$$\sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

maximaal is.

Invoer

De invoer is een geheel getal $n \in [2, 8]$.

Bonus: Indien je programma ook elke $n \in [2, 100]$ aankan binnen 10s op onze testmachine, wordt 60 minuten van je tijd afgetrokken in de rangschikking.

Uitvoer

De uitvoer is een rij van gehele getallen gescheiden door spaties en afgesloten door een newline.

Voorbeelden

Invoer:

2

Uitvoer:

1 2

Invoer:

3

Uitvoer:

2 1 3

Invoer:

4

Uitvoer:

2 4 1 3

Invoer:

5

Uitvoer:

2 4 1 5 3

Go

Een Go bord is een vierkant rooster, waarop twee spelers om beurten een steen plaatsen. De ene speler speelt met wit en de andere met zwart. Een *groep* (stenen) is een aantal stenen op het bord van dezelfde kleur en die allemaal met elkaar verbonden zijn via verticale en horizontale overgangen. Op het volgende 9x9 bord heeft zwart (z) een groep van 4 stenen en een groep van 2 stenen, terwijl wit (w) groepen heeft van 6, 4 en 1 ste(en).

```
- - - - - - - - -
- - - - - - - - -
- - w w - - - - -
- w z w - - - - -
- w z w w - - - - -
- w z w - - - - -
- w z - - - - - -
- z w - - - - - -
- z - - - - - - -
```

Het aantal vrijheden van een groep is het aantal lege plaatsen dat (opnieuw horizontaal en verticaal) aan deze groep grenst. Zwarts groep van 4 stenen heeft bijvoorbeeld 1 vrijheid, terwijl zijn groep van twee stenen er 3 heeft. Als een speler een zet doet waardoor een groep van zijn tegenstander zijn laatste vrijheid kwijtraakt, dan wordt deze laatste groep door hem gevangen genomen en verdwijnt ze van het speelbord.

Probleem

Bepaal het effect van een zet van wit op het speelbord.

Invoer

De invoer bestaat uit een 9x9 speelbord gevuld door de positie van de volgende zet die door de witte speler wordt uitgevoerd.

De eerste 9 invoerlijnen zijn de 9 rijen (van boven naar onder) van het bord. Elke lijn bestaat uit 9 karakters die van links naar rechts de posities in de rij weergeven. Elke positie wordt voorgesteld door een karakter:

- - is een vrije positie

- z is ingenomen door een zwarte steen
- w is ingenomen door een witte steen

Merk op dat we bij de voorbeelden spaties tussen de kolommen gezet hebben voor de leesbaarheid . Die spaties komen niet voor in de effectieve invoer.

De positie van de volgende zet wordt gegeven door de laatste invoerlijn, van de vorm:

rijnummer kolomnummer

waarbij geteld wordt van boven naar beneden en links naar rechts, bv:

7 4

Uitvoer

De gevraagde uitvoer is het resulterende 9x9 spelbord, in het zelfde formaat als in de invoer.

Voorbeelden

Invoer:

```

- - - - -
- - - - -
- - w w - - -
- w z w - - -
- w z w w - - -
- w z w - - -
- w z - - -
- z w - - -
- z - - -
7 4

```

Uitvoer:

```

- - - - -
- - - - -
- - w w - - -
- w - w - - -
- w - w w - - -
- w - w - - -
- w - w - - -
- z w - - -
- z - - -

```

Invoer:

```

- - - w w - - -
- - w z w - w - -
- - w z w w z w -
- - w z w z z z w
- - w z - z w w -
- w z w z z w - -
- w z z w z w - -
- - w z z z w - -
- - - w w w - - -
5 5

```

Uitvoer:

```

- - - w w - - -
- - w - w - w - -
- - w - w w - w -
- - w - w - - w
- - w - w - w w -
- w - w - - w - -
- w - - w - w - -
- - w - - w - -
- - - w w w - - -

```


Deuren

Je werkt voor een excentrieke architect. De architect gaat bij het plaatsen van de muren totaal willekeurig te werk. Nadien weet hij niet hoeveel kamers zijn ontwerpen tellen. Jouw taak bestaat erin te bepalen hoeveel deuren nodig zijn om alle kamers te kunnen bereiken van buitenaf.

Probleem

Elk grondplan van de architect bestaat uit een rooster van 11 op 11 punten. De coördinaten van de punten kunnen dus uitgedrukt worden met de getallen van 0 tot en met 10. Tussen deze punten kunnen er enkel horizontale en verticale verbindingen, muren, bestaan. Alles speelt zich af op één enkel verdiep.

Een deur is een verbinding tussen twee naast elkaar gelegen punten. De twee punten maken op het grondplan deel uit van een muur. Een deur laat toe om van de ene aanpalende ruimte naar de andere te gaan, en vice versa.

Bepaal het minimum aantal deuren om alle ruimtes te kunnen bereiken van buitenaf.

Invoer

De eerste lijn van de invoer is het aantal muursegmenten n (verbindingen tussen twee verticaal of horizontaal aangrenzende punten) waaruit de constructie bestaat. Vervolgens komen er n lijnen met elk 4 getallen op: $x_1 y_1 x_2 y_2$. Deze geven de coördinaten van het begin- en eindpunt van het muursegment. Voor al deze getallen geldt $0 \leq x \leq 10$. Er zijn maximaal 100 muursegmenten in een constructie.

Uitvoer

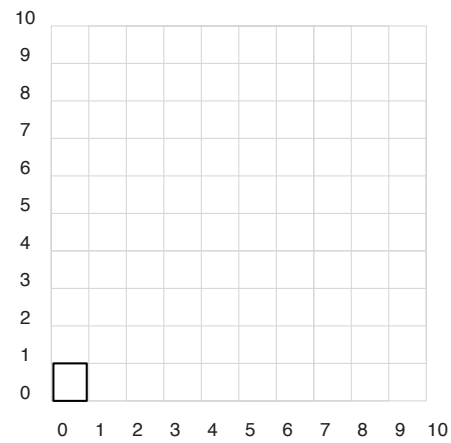
De uitvoer bestaat uit een lijn met 1 getal op: het minimaal aantal nodige deuren om alle kamers te kunnen bereiken van buitenaf.

Voorbeelden

Invoer:

```
4
0 0 0 1
0 1 1 1
1 1 1 0
1 0 0 0
```

Afbeelding:



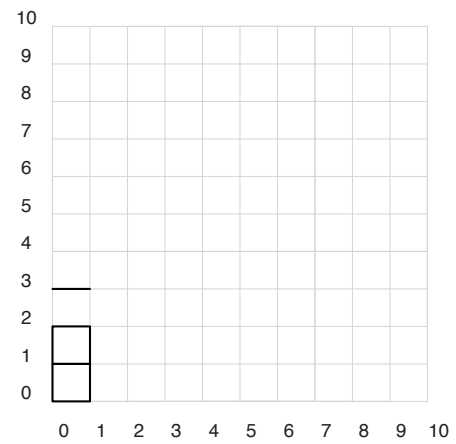
Uitvoer:

1

Invoer:

```
8
0 0 0 1
0 1 1 1
1 1 1 0
1 0 0 0
1 1 2 1
2 1 2 0
2 0 1 0
3 0 3 1
```

Afbeelding:



Uitvoer:

2