Opdracht 0. Flat-puzzel

In veel puzzeltijdschriften kun je zogenaamde flatpuzzels aantreffen.
Deze bestaan uit een vierkant diagram van n bij n hokjes. In iedere rij en iedere kolom moeten de getallen 1 tot en met n elk één keer worden ingevuld. Deze staan voor flats van verschillende hoogte. Aan weerszijden van de rij of kolom staan getallen die aangeven hoeveel flats vanaf die kant zichtbaar zijn. (Hoe hoger het getal, hoe hoger de flat; een hogere flat blokkeert het zicht op een lagere erachter).

Door logisch redeneren kun je erachter komen welke flat waar staat.

	3	1	2	2	
2					2
2					1
3					2
1					3
	1	3	2	2	

Voorbeeld diagram flatpuzzel van 4 bij 4.

In deze opgaven ga je in drie stappen deze puzzels oplossen.

Invoer:

Invoer bij alle opdrachten is een bestand **`flats.in`** dat bestaat uit vijf regels. Op de eerste regel staat een getal n (2 < n < 10) dat aangeeft hoeveel flats er in iedere rij staan.
De tweede tot en met vijfde regel bestaan elk uit n getallen, gescheiden door spaties.
De tweede regel geeft de informatie boven het diagram, van links naar rechts.
De derde regel geeft de informatie rechts van het diagram, van boven naar beneden.
De vierde regel geeft de informatie onder het diagram, van links naar rechts.
De vijfde regel geeft de informatie links van het diagram, van boven naar beneden.

Voorbeeld bestand **`flats0.in`**:

```
4
3 1 2 2
2 1 2 3
1 3 2 2
2 2 3 1
```

Dit bestand wordt bij de voorbeelden bij de opgaven als invoer gebruikt.

Voorbeeldbestanden en testen:

Er zijn bestanden **`flats0.in`**, **`flats1.in`** tot en met **`flats5.in`** beschikbaar waarmee je je programma kunt uitproberen.
Er is een batchfile **`test01.bat`**, die je kunt gebruiken op de volgende manier:

```
test01 nio0a flats0.in
```

Met deze opdracht test je het programma **`nio0a`** (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit **`flats0.in`** (of op deze plaats één van de andere bestanden) naar het bestand **`flats.in`** wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 0. Taakoverzicht.

Opdracht	Programma	Invoer	Uitvoer	Tijd-limiet	Testen	Punten per test	Totaal
0A	nio0a	flats.in	0a.uit	1 s	5	4	20
0B	nio0b	flats.in	0b.uit	2 s	5	7	35
0C	nio0c	flats.in	0c.uit	3 s	5	9	45

Opdracht 0A: Wat is meteen duidelijk?

Schrijf een programma **nio0a** dat een bestand **flats.in** inleest. Uitvoer is een bestand **0a.uit** van n regels van n tekens. Er staat het getal n als op de overeenkomstige positie een flat van hoogte n moet staan, en dat al bekend is uit de informatie over de randen, anders een 0.

Voorbeeld:

```
0400
0004
0000
4000
```

(weliswaar weet je nu ook de vierde flat van hoogte n, maar die is nog niet meteen bekend en wordt niet in je uitvoer aangegeven.)

Opdracht 0B: Alle mogelijkheden voor n en n-1

Schrijf een programma **nio0b** dat een bestand **flats.in** inleest. Uitvoer is een bestand **0b.uit** van 2n regel van elk n tekens.
Op de eerste n regels staat voor iedere positie een n als er een flat met hoogte n kan staan, een 0 als dat niet het geval is. Dan volgen n regels met een n-1 of een 0. Je moet alle informatie die je al hebt gebruiken!

Voorbeeld:

```
0400
0004
0040
4000
0003
3000
0300
0030
```

In dit voorbeeld weet je alle posities van 3 en 4 al. Dat hoeft niet het geval te zijn!

Opdracht 0C: Oplossen

Schrijf een programma **nio0c** dat een bestand **flats.in** inleest. Uitvoer is een bestand **0c.uit** van n regel van elk n tekens. Het bestand geeft een weergave van de goede oplossing van de flatpuzzel.

Voorbeeld:

```
1423
3214
2341
4132
```