

Opdracht 1. Stoelen reserveren

Je bent beheerder van een theater en je moet een systeem maken om de stoelreserveringen te kunnen verwerken. Daarom ga je een paar computerprogramma's schrijven die je daarbij kunnen helpen.

De stoelen staan opgesteld in een rechthoek; er zijn r rijen ($0 < r < 40$) en elke rij bestaat weer uit s stoelen ($0 < s < 30$). De stoelen zijn genummerd van 1 tot en met $r \times s$. De achterste rij bestaat uit de stoelen 1 tot en met s .

Achter	1	2	3	4	5
	6	7	8	9	10
	11	12	13	14	15
Voor	16	17	18	19	20

Voorbeeld van een zaal met 20 stoelen (4 rijen van 5 stoelen)

Bij iedere bestelling geeft je systeem alleen maar kaartjes naast elkaar uit. In het voorbeeld hierboven kunnen dus nooit meer dan 5 kaartjes tegelijk worden besteld.

Invoer:

Invoer bij alle opdrachten is een bestand `theater.in` dat bestaat uit een onbekend aantal regels. Op de eerste regel staat een getal r dat aangeeft hoeveel rijen stoelen er in het theater zijn. Op de tweede regel een getal s dat aangeeft hoeveel stoelen er in iedere rij staan. Daarna komen er regels waarop de opeenvolgende bestellingen van kaartjes staan aangegeven. Op iedere regel staat een getal k dat aangeeft hoeveel kaartjes er worden besteld; $0 \leq k \leq s$. De laatste invoerregel bevat het getal 0.

Voorbeeld bestand `theater0.in`:

```
4
5
3
5
2
4
3
4
1
0
```

(hieronder worden deze bestellingen aangeduid met a, b, c, d, e, f en g)

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

Voorbeeldbestanden en testen:

Er zijn bestanden `theater0.in`, `theater1.in` tot en met `theater5.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test1.bat`, die je kunt gebruiken op de volgende manier:

```
test1 niola theater0.in
```

Met deze opdracht test je het programma `niola` (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit `theater0.in` (of op deze plaats één van de andere bestanden) naar het bestand `theater.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 1. Taakoverzicht.

Opdracht	Programma	Invoer	Uitvoer	Tijdelimiet	Testen	Punten per test	Totaal
1A	nio1a	theater.in	1a.uit	1 s	5	2	10
1B	nio1b	theater.in	1b.uit	1 s	5	3	15
1C	nio1c	theater.in	1c.uit	1 s	5	4	20
1D	nio1d	theater.in	1d.uit	1 s	5	4	20
1E	nio1e	theater.in	1e.uit	5 s	5	7	35

Opdracht 1A: Aantal stoelen

Schrijf een programma `nio1a` dat een bestand `theater.in` inleest. Uitvoer is een bestand `1a.uit` van één regel. Hierop staat het aantal stoelen van het theater aangegeven.

Voorbeeld: 20

Opdracht 1B: Van achteren en dan van links naar rechts

Schrijf een programma `nio1b` dat een bestand `theater.in` inleest. Uitvoer is een bestand `1b.uit` van r regel van elk s tekens. Ieder teken stelt één stoel voor; er staat een 1 als de stoel gereserveerd is en een 0 als dat niet het geval is (zie het voorbeeld hieronder).

Het programma moet het volgende voorschrift toepassen:

- Neem de achterste rij met voldoende vrije stoelen naast elkaar.
- Wijs op die rij de laagst genummerde vrije plaatsen toe.
- Als er niet genoeg vrije plaatsen naast elkaar zijn worden er helemaal geen plaatsen gereserveerd.

Voorbeeld:

11111	Achter	a	a	a	c	c
11111		b	b	b	b	b
11111		d	d	d	d	g
11100	Voor	e	e	e	19	20

Opdracht 1C: Altijd een zo leeg mogelijke rij vullen

Schrijf een programma `nio1c` dat een bestand `theater.in` inleest. Uitvoer is een bestand `1c.uit` van r regel van elk s tekens. Ieder teken stelt één stoel voor; er staat een 1 als de stoel gereserveerd is en een 0 als dat niet het geval is (zie het voorbeeld hieronder).

Het programma moet het volgende voorschrift toepassen:

- Bepaal de rij met de meeste vrije plaatsen. Als het er meer zijn de achterste ervan.
- Wijs op die rij de laagst genummerde vrije plaatsen toe.
- Als er niet genoeg vrije plaatsen naast elkaar zijn worden er helemaal geen plaatsen gereserveerd.

Voorbeeld:

11110	Achter	a	a	a	g	5
11111		b	b	b	b	b
11111		c	c	e	e	e
11110	Voor	d	d	d	d	20

Opdracht 1D: Altijd een zo vol mogelijke rij vullen

Schrijf een programma `nio1d` dat een bestand `theater.in` inleest. Uitvoer is een bestand `1d.uit` van `r` regel van elk `s` tekens. Ieder teken stelt één stoel voor; er staat een 1 als de stoel gereserveerd is en een 0 als dat niet het geval is (zie het voorbeeld hieronder).

Het programma moet het volgende voorschrift toepassen:

- Bepaal de rij met de minste vrije plaatsen; het moeten er natuurlijk wel genoeg zijn. Als er meer zijn de achterste ervan.
- Wijs op die rij de laagst genummerde vrije plaatsen toe.
- Als er niet genoeg vrije plaatsen naast elkaar zijn worden er helemaal geen plaatsen gereserveerd.

Voorbeeld:	11111	Achter	a	a	a	c	c
	11111		b	b	b	b	b
	11111		d	d	d	d	g
	11100	Voor	e	e	e	19	20

Opdracht 1E: Achteraf pas plannen

Schrijf een programma `nio1e` dat een bestand `theater.in` inleest. Uitvoer is een bestand `1e.uit` van `r` regel van elk `s` tekens. Ieder teken stelt één stoel voor; er staat een 1 als de stoel gereserveerd is en een 0 als dat niet het geval is (zie het voorbeeld hieronder).

Het programma moet het volgende voorschrift toepassen:

- Bekijk alle binnengekomen reserveringen.
- Vul de zaal zo veel mogelijk.

Voorbeeld:	11111	Achter	a	a	a	c	c
	11111		b	b	b	b	b
	11111		d	d	d	d	g
	11110	Voor	f	f	f	f	20

Bij grotere invoer gaat dit heel erg veel tijd kosten. Daarom geldt het volgende:

Als je programma de beste uitvoer van de programma's van onderdeel 1B, 1C en 1D geeft krijg je tenminste 1 punt.

Als je programma het theoretisch maximum aan stoelen weet te plaatsen krijg je 7 punten.

Als je programma een tussenliggend aantal bereikt krijg je ook een tussenliggend aantal punten.

Als je programma een aantal stoelen aangeeft dat niet overeenkomt met de invoer krijg je geen punten!