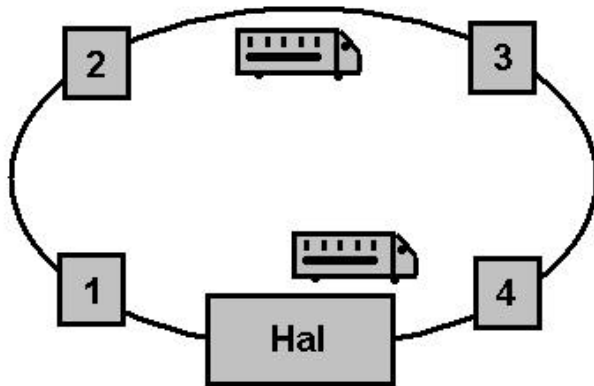


## Opgave 2. De vliegveldbussen.

Op een vliegveld zijn de afstanden zo groot dat passagiers met bussen worden vervoerd. Er is één centrale hal, vanwaar altijd veel mensen willen vertrekken, en er zijn  $N$  gebouwen, waarbij  $N$  een even getal is, met  $2 = N = 8$ .



De afdeling planning van de luchthaven heeft voor ieder tijdstip een goed beeld van de aantallen passagiers die van het ene gebouw naar het andere willen reizen. Hieronder zie je een voorbeeld van dergelijke informatie voor een vliegveld met 4 gebouwen.

Vervoerswensen tussen de verschillende gebouwen van de luchthaven.					
Van \ Naar	Hal	Gebouw 1	Gebouw 2	Gebouw 3	Gebouw 4
Hal	0	100	150	200	250
Gebouw 1	140	0	30	40	50
Gebouw 2	190	60	0	80	70
Gebouw 3	160	55	45	0	65
Gebouw 4	210	25	75	35	0

Links in dit schema staan de locaties waar de passagiers zich bevinden. Bovenaan in het schema staan de plaatsen waar ze naar vervoerd willen worden. Zo zijn er 80 passagiers die van Gebouw 2 naar Gebouw 3 willen, en 45 passagiers die van Gebouw 3 naar Gebouw 2 willen.

De bussen rijden slechts één rondje. Ze beginnen en eindigen bij de Hal en stoppen bij de gebouwen in de volgorde waarin ze genummerd zijn (de oplopende bussen; in het plaatje is dat met de klok mee), of juist in aflopende volgorde (in het plaatje tegen de klok in). **Overstappen is niet mogelijk.**

In deze opgave ga je met deze gegevens een aantal zaken uitrekenen.

Invoer.

Invoer is een bestand `wensen.in` met op de eerste regel een getal  $N$ . Vervolgens zijn er  $N+1$  regels met op elke regel  $N+1$  getallen, gescheiden door een spatie.

Voorbeeld: Bestand `wensen0.in`

```
4
0 100 150 200 250
140 0 30 40 50
190 60 0 80 70
160 55 45 0 65
210 25 75 35 0
```

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

## Voorbeeldbestanden en testen:

Er zijn bestanden `wensen0.in`, `wensen1.in` tot en met `wensen5.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test2.bat`, die je kunt gebruiken op de volgende manier:

```
test2 nio2a wensen0.in
```

Met deze opdracht test je het programma `nio2a` (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit `wensen0.in` (of op deze plaats één van de andere bestanden) naar het bestand `wensen.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

## Opgave 2 overzicht

Onderdeel	Programma	Uitvoer	Tijdlimiet per test	Aantal testen	Punten per test	Totaal
2A	<code>nio2a</code>	<code>2a.uit</code>	1 sec	5	2	10
2B	<code>nio2b</code>	<code>2b.uit</code>	1 sec	5	3	15
2C	<code>nio2c</code>	<code>2c.uit</code>	1 sec	5	5	25
2D	<code>nio2d</code>	<code>2d.uit</code>	5 sec	5	10	50

### Onderdeel 2A: Aantal passagiers

Schrijf een programma `nio2a` dat een bestand `wensen.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2a.uit` dat bestaat uit één regel. Daarop staat het totale aantal passagiers dat vervoerd wil worden aangegeven.

Uitvoer bij het gegeven voorbeeld:      2030

### Onderdeel 2B: Per gebouw

Schrijf een programma `nio2b` dat een bestand `wensen.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2b.uit` dat bestaat uit  $N+1$  regels van elk twee getallen, gescheiden door een spatie. Op iedere regel staat eerst aangegeven hoeveel passagiers bij een gebouw willen vertrekken, daarna het aantal passagiers dat er wil aankomen. De eerste regel in de uitvoer heeft betrekking op de hal, daarna komen er  $N$  regels met de gegevens van de gebouwen 1 tot en met  $N$ .

Uitvoer bij het gegeven voorbeeld:      700 700  
   260 240  
   400 300  
   325 355  
   345 435

## Onderdeel 2C: De kortste weg

Voor passagiers die van het ene gebouw naar het andere willen, ligt vast welke bus zet moeten hebben: Van Gebouw 1 naar Gebouw 4 neem je de oplopende bus, van Gebouw 4 naar Gebouw 1 de aflopende (bedenk dat overstappen niet mogelijk is!). Passagiers die vanuit de Hal vertrekken of er naar toe willen reizen kunnen kiezen in welke richting ze dat doen. In deze opgave gaan we er van uit dat ze kiezen voor een route met zo weinig mogelijk stops.

Schrijf een programma `nio2c` dat een bestand `wensen.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2c.uit` dat bestaat uit twee regels. Op de eerste regel staat het maximum aantal passagiers voor de bussen die de oplopende route volgen, op de tweede regel het maximum aantal passagiers voor de bussen die de aflopende route volgen.

Uitvoer bij het gegeven voorbeeld:

```
370
450
```

## Onderdeel 2D: Zo weinig mogelijk bussen

De afdeling Planning van het vliegveld wil de beschikbare bussen nu zo inzetten dat er zo weinig mogelijk bussen nodig zijn. In elke bus kunnen 60 passagiers meereizen. Bij het voorbeeld van onderdeel 2C zouden 7 bussen voor de oplopende route en 8 voor de aflopende route nodig zijn.

Schrijf een programma `nio2d` dat een bestand `wensen.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2d.uit` dat bestaat uit  $N$  regels van elk twee getallen, gescheiden door een spatie. Op iedere regel  $G$  staat eerst aangegeven hoeveel passagiers voor Gebouw  $G$  vanuit de Hal moeten vertrekken met de oplopende bussen, daarna het aantal passagiers bij Gebouw  $G$  voor de Hal dat moet vertrekken met de aflopende bussen. De gegeven oplossing moet de inzet van een minimaal aantal bussen mogelijk maken. In veel gevallen zal meer dan één indeling mogelijk zijn; je programma dient maar één indeling te geven.

Uitvoer bij het gegeven voorbeeld:

```
100 0
150 0
90 150
0 210
```

Met deze verdeling zijn in beide richtingen 6 bussen nodig. Met minder dan 12 bussen in totaal is geen oplossing mogelijk. Op regel 3 staat hier dat 90 passagiers vanuit de hal richting Gebouw 3 vertrekken met de oplopende bussen, en 150 passagiers vanuit Gebouw 3 naar de hal vertrekken met de aflopende bussen. De aantallen voor de aflopende bussen staan hier niet aangegeven; deze volgen uit de invoer!