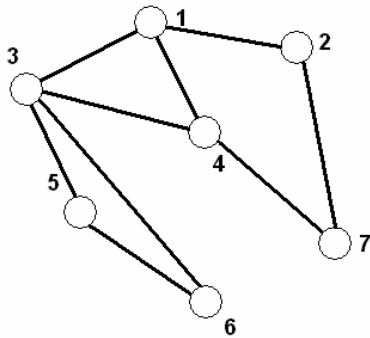


Opdracht 2. Stadsplattegrond.

Op een plattegrond staan gegevens van de pleinen en verbindingswegen van een stad. De pleinen van de stad zijn genummerd van 1 tot en met P . Er zijn W verbindingswegen. Op ieder plein komen tenminste twee wegen uit. Tussen twee pleinen loopt hoogstens één verbindingsweg. Bij deze plattegrond moet je een aantal dingen uitzoeken.

Voorbeeld plattegrond:



Invoer:

Invoer is een bestand `kaart.in`.

Op de eerste regel staat een getal P , met $3 \leq P \leq 200$, dat het aantal pleinen op de plattegrond aangeeft. Op de tweede regel staat een getal W , met $6 \leq W \leq 6000$, dat het aantal verbindingswegen op de plattegrond aangeeft. Vervolgens zijn er W regels, met op iedere regel 2 pleinnummers V en N , gescheiden door een spatie. Zo'n regel geeft aan dat er een weg loopt van plein nummer V naar plein nummer N .

Voorbeeld bestand `kaart0.in`:

```
7
9
1 2
1 3
1 4
2 7
3 4
3 5
3 6
4 7
5 6
```

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

Voorbeeldbestanden en testen:

Er zijn bestanden `kaart0.in`, `kaart1.in` tot en met `kaart5.in` beschikbaar waarmee je je programma kunt uitproberen.

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

```
test2 nio2a kaart0.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 `kaart0.in` (of op deze plaats één van de andere bestanden) naar het bestand `kaart.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 2. Taakoverzicht.

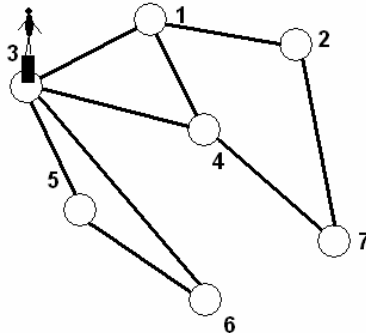
Onderdeel	Programma	Uitvoer	Tijdlimiet per test	Aantal testen	Punten per test	Totaal
2A	nio2a	2a.uit	2 sec	5	2	10
2B	nio2b	2b.uit	2 sec	5	8	40
2C	nio2c	2c.uit	n.v.t.	1	20	20
2D	nio2d	2d.uit	2 sec	5	6	30

Opdracht 2A: Het standbeeld.

Het gemeentebestuur wil graag een standbeeld plaatsen op één van de pleinen waar de meeste wegen uitkomen. Als er meer pleinen zijn met hetzelfde aantal toegangswegen kiest men het laagst genummerde.

Schrijf een programma `nio2a` dat een bestand `kaart.in` inleest. Uitvoer is een bestand `2a.uit` van één regel; hierop staat het nummer aangegeven van het plein waarop het standbeeld komt.

Voorbeeld: 3



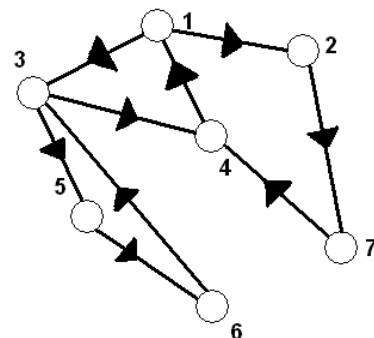
Opdracht 2B: Eenrichtingsverkeer.

Omdat het vaak nogal druk is op de verbindingswegen besluit het gemeentebestuur om van alle wegen eenrichtingsverkeer te maken. Daarbij is het uiteraard wel nodig dat alle pleinen bereikbaar blijven!

Bij alle plattegronden waarmee je programma wordt getest is het mogelijk een goed systeem voor eenrichtingsverkeer te maken!

Schrijf een programma `nio2b` dat een bestand `kaart.in` inleest. Uitvoer is een bestand `2b.uit` dat bestaat uit W regels; hierop staat voor alle wegen (in de volgorde van het invoerbestand) een + als de richting van V naar R wordt, en een - als het andersom wordt.

Voorbeeld:
 +
 +
 -
 +
 +
 +
 -
 -
 +



Opdracht 2C: Een onmogelijke plattegrond.

In de inleiding van de hele opgave staan specificaties voor de plattegrond.

In de inleiding van onderdeel 2B staat aangegeven dat, bij alle invoer die je programma krijgt aangeboden, een goed eenrichtingsverkeer systeem mogelijk is.

Kun jij een plattegrond laten zien die voldoet aan de specificaties uit het begin van de opgave, maar waarvoor geen eenrichtingsverkeer systeem mogelijk is waarbij vanuit ieder plein ieder ander plein te bereiken is? En probeer dat voorbeeld zo klein mogelijk te houden!

Schrijf een BESTAND `nio2c.uit` dat een plattegrond beschrijft op de manier zoals het bestand `kaart.in` dat je bij andere onderdelen als invoer krijgt. Voor een goede plattegrond krijg je in ieder geval 10 punten; met een minimaal aantal wegen kun je nog eens 10 punten verdienen.

Opdracht 2D: Een controleprogramma.

Natuurlijk is het gemeentebestuur heel blij met alle programma's die ze kunnen gebruiken om eenrichtingsverkeer te introduceren. Men wil echter ook wel graag kunnen controleren of de resultaten van deze programma's correct zijn.

Schrijf een programma `nio2d` dat een bestand `kaart.in` inleest en een bestand `2b.uit` (uiteraard horen deze bestanden bij elkaar en beschrijven ze dezelfde plattegrond). Uitvoer is een bestand `2d.uit` van één regel. Als vanuit iedere plein ieder ander plein kan worden bereikt staat er 'OK.'; als dat niet het geval is, is er dus een plein P van waaruit een ander plein A niet bereikbaar is. Dan staat op de regel uitvoer de getallen P en A, gescheiden door een spatie. Als meerdere pleinen onderling onbereikbaar zijn geworden moet je programma er toch maar één weergeven!

Omdat het al te makkelijk zou zijn om voor alle zekerheid aan te geven dat alle pleinen onderling bereikbaar zijn, wordt je programma getest met (vijf) groepen van drie testgevallen. Alleen als alle drie testgevallen goed worden opgelost krijgt je programma punten voor die test.

Voorbeeld :

Invoer	Testgeval 1	Testgeval 2	Testgeval 3
7			
8			
1 2	+	+	+
1 3	+	+	+
1 4	-	-	+
2 7	+	+	+
3 4	+	-	+
3 5	+	+	+
3 6	-	-	+
4 7	-	-	+
5 6	+	+	+
Resultaat	OK.	3 4	2 1