

Nederlands Kampioenschap Programmeren 1991

Opgave A - De magische bollen

De grote tovenaar Merlijn heeft voor het eerst sinds tijden een probleem waar hij niet zo snel raad mee weet. Zijn vier magische bollen zijn alleen actief wanneer ze alle vier een rode gloed hebben. Helaas, dit is niet altijd het geval, integendeel ! 's Morgens wanneer hij wakker wordt heeft elke bol een rode, een groene, een gele of een blauwe gloed. Wanneer Merlijn naar een van de bollen wijst, wijzigt de gloed hiervan. De volgorde waarin de kleuren elkaar opvolgen is niet elke dag dezelfde, maar deze volgorde is bij elke bol wel gelijk. Zo kan het gebeuren dat de bollen de ene dag in de volgorde rood-blauw-geel-groen van kleur veranderen en de andere dag in de volgorde blauw-rood-groen-geel. Wanneer Merlijn naar een bol wijst (waardoor deze bol dus van kleur verandert) en daarna nogmaals naar dezelfde bol wijst, dan blijft de kleur van deze bol ongewijzigd. Hij zal dus eerst naar een andere bol moeten wijzen. Merlijn wil nu een programma hebben dat, bij een gegeven beginsituatie (de kleuren van de bollen) en bij een gegeven volgorde waarin de kleuren elkaar opvolgen, een snelste manier geeft waarmee hij alle bollen een rode gloed kan bezorgen.

Op de eerste regel van de invoer staat het aantal runs. Elke run bestaat uit twee regels : de eerste regel bevat de vier begin-kleuren van de bollen en op de tweede regel staat de kleur-volgorde waarin de bollen veranderen (de volgorde bevat de vier verschillende kleuren).

De uitvoer bevat voor elke run (op een eigen regel) het minimale aantal bollen dat moet worden aangewezen om alle bollen rood te maken. Daarnaast moet een volgorde gegeven worden waarin de bollen aangewezen worden om ze in dit minimale aantal stappen alle vier rood te laten worden. Het aantal en de volgorde moeten gescheiden worden door een dubbele punt (:). Verder moeten alle aantallen (en de aantallen en de dubbele punt) gescheiden worden door spaties.

Voorbeeldinvoer

```
2
rood groen groen geel
rood geel blauw groen
rood rood rood blauw
rood geel blauw groen
```

Uitvoer bij de voorbeeldinvoer

```
5 : 4 2 4 3 4
10 : 1 4 2 4 1 2 1 2 1 2
```

Nederlands Kampioenschap Programmeren 1991

Opgave B - DJ "Koker"

DJ "Koker" is een vreemde vogel. Hij is een uitstekende platendraaier, hij kan de plaatjes lekker aan elkaar praten en hij is van tijd tot tijd ook goed voor "een strakke mix". Hij heeft echter n eigenaardigheid; het is nog nooit voorgekomen dat hij twee platen achter elkaar heeft gedraaid zonder dat er in de titel van deze platen een woord van overeenkomst zat (en hij is ook niet van plan om van deze gewoonte af te wijken). Het spreekt voor zich dat het een heidens karwei is om steeds weer uit een aantal verzoeknummers een maximale serie platen te halen zodanig dat "de koker" niet breekt met zijn traditie. Hij wil dus jullie hulp inroepen voor dit probleem.

Er wordt gevraagd om een programma te schrijven dat uitgaande van een groep nummers bepaalt hoelang een "koker-serie" maximaal kan duren. Het is niet toegestaan om een nummer meer dan n keer te draaien. Het kan wel voorkomen dat twee nummers uit een groep dezelfde titel hebben. Daarnaast is het een gegeven dat de eerste plaat uit de groep al bij voorbaat op de draaitafel wordt gelegd; de andere platen moeten hier op aansluiten. Twee nummers mogen alleen na elkaar gedraaid worden wanneer er **hele** woorden uit de titels overeenkomen (Kenny G's "songbird" en "your song" van Elton John voldoen dus niet aan de gestelde eisen).

De invoer bevat een aantal runs. Op de eerste regel van een run staat het aantal platen in een groep (maximaal 10). Elk nummer van een groep neemt twee regels in beslag. Op de eerste regel staat de titel van het nummer. Een titel bevat maximaal 20 woorden die gescheiden zijn door n spatie. De woorden (lengte ≤ 15) bestaan uit uitsluitend kleine letters (a...z). V r het eerste woord en n het laatste woord staan geen spaties. Op de tweede regel staat de duur van het nummer in hele seconden (maximaal 500 seconden). Na de laatste groep nummers volgt een regel met het getal 0. De invoerfile bevat geen lege regels (behalve na de laatste 0).

De uitvoer bevat voor elke run (op een eigen regel) de maximale tijdsduur van een koker-serie.

Voorbeeldinvoer

```
5
een nacht alleen
210
een kopje koffie
240
alleen met jou
220
even aan mijn moeder vragen
300
belgie
275
5
the year of the cat
375
moments in love
300
in between days
260
in the air tonight
```

180
sowing the seeds of love
320
0

Uitvoer bij de voorbeeldinvoer

450
1435

Bij het testen van het programma wordt er een tijds-tolerantie van **vier** minuten gehanteerd.

eerste regel een getal dat aangeeft hoeveel test-cases (runs) er volgen. Elke test-case beslaat drie regels. De eerste daarvan bevat een programma-streng, de tweede regel bevat een virus-streng en de derde regel bevat een blokkeer-streng. Elke streng bevat maximaal 2000 bits. Er komen geen spaties en geen lege regels voor in de invoer. Dit betekent dat elke voorkomende streng minstens n bit bevat. Ook de laatste streng wordt afgesloten met een EOLN-teken.

De uitvoer die uw programma produceert bestaat uit n regel voor iedere case. Op die regel staat een van de volgende boodschappen:

Niet selectief

Besmet

Niet besmet

"Niet selectief" wanneer een bit gewijzigd wordt dat niet in het besmet gebied staat.

"Besmet" als er uitsluitend bits gewijzigd worden uit besmet gebied.

"Niet besmet" als de programma-streng geen virussen bevat.

Voorbeeldinvoer

```
2
010101110101000
0101
1010
000000000000010
00
1
```

Uitvoer bij de voorbeeldinvoer

Besmet

Niet selectief

Nederlands Kampioenschap Programmeren 1991

Opgave D - Minotaurus

In het paleis van koning Minos op Knossos zijn een aantal gouden zwaarden verborgen. De held van dit verhaal wil in het geniep zoveel mogelijk van deze zwaarden wegnemen. Hij heeft een kaart te pakken gekregen met daarop de vindplaatsen van deze zwaarden, een veilige ingang en een veilige uitgang van het paleis. Bovendien bezit hij een gouden sleutel waarmee alle deuren in het paleis geopend kunnen worden. Help onze held door een programma te schrijven dat aangeeft wat het minimale aantal deuren is dat geopend moet worden om de zwaarden uit het paleis mee te nemen.

Het paleis bestaat uit een aantal vierkante vertrekken die tezamen een soort van (rechthoekig) schaakbord vormen. Soms kan men ongehinderd van het ene vertrek naar het andere lopen, soms zijn de vertrekken gescheiden door een muur, en op andere plaatsen zijn de vertrekken verbonden door een deur. Onze held kan deze deuren openen. Om het gevaar van ontdekking zo klein mogelijk te houden wil hij dit wel zo weinig mogelijk doen.

Het paleis wordt soms bewaakt door de legendarische minotaurus: half mens, half stier. Deze doolt door een aantal vertrekken, maar kan met zijn hoeven geen deuren open maken. Onze held dient er voor te waken dat hij geen vertrek betreedt waar de minotaurus kan komen. Dit betekent dat sommige zwaarden, en soms zelfs de uitgang, voor hem onbereikbaar zijn.

Een kaart in de invoer wordt beschreven met behulp van een aantal tekens. Gebruikt worden '+' voor de hoekpunten van de vertrekken, '-' en '|' voor (ondoordringbare) muren, en '' voor deuren die met behulp van de gouden sleutel geopend kunnen worden. 'I', 'U', 'Z' en 'M' geven de plaatsen aan van de ingang, de uitgang, de zwaarden, en de minotaurus. Zowel lege vertrekken, als verbindingen tussen vertrekken zonder muur of deur, worden met een spatie aangegeven. Zie het voorbeeld.

De invoer bevat een aantal van deze kaarten. Elke kaart wordt vooraf gegaan door twee regels met daarop het aantal vertrekken horizontaal, resp. vertikaal. De invoer wordt afgesloten met een 'triviale' kaart, dwz. drie regels met daarop 0, 0, + (dit is de kaart van een paleis van nul bij nul vertrekken).

De uitvoer dient voor elke kaart uit de invoer (behalve de afsluitende) de volgende informatie te bevatten. Indien de uitgang onbereikbaar is, een regel met de tekst 'UITGANG ONBEREIKBAAR'. Anders twee regels, een met daarop het aantal bereikbare zwaarden ('x ZWAARDEN BEREIKBAAR') en een met het minimale aantal te openen deuren ('y DEUREN OPENEN') om deze zwaarden naar buiten te brengen. Besluit elke oplossing met een regel met daarop '=== '.

NB. Elke kaart (behalve de afsluitende) bevat tussen de nul en acht zwaarden, precies n ingang en uitgang, en ten hoogste n minotaur. De minotaur kan de ingang niet bereiken. De buitenste rand van het paleis bestaat uit muren (en hoekpunten). Toegang alleen via de aangegeven vertrekken. Het paleis is niet groter dan 50x50 vertrekken.

NB. De held dient dus een wandeling te beschrijven van ingang naar uitgang, via de bereikbare zwaarden. Deuren mogen meerdere malen gepasseerd worden, waarbij ELKE passage telt als het

openen van een deur, zelfs indien dezelfde deur direct na elkaar gebruikt wordt om een vertrek in en uit te gaan.

Voorbeeldinvoer

```
7
5
+--+--+--+--+--+--+--+
| I  "          Z |
+--+ +--+--+--+--+--+--+
| "  "          |
+ + +--+--+--+--+--+--+
| Z | |          |
+--+ + +--+--+ + +
| " | M | |      |
+ + + + + + + + +
| | | | U"      |
+--+--+--+--+--+--+--+
9
3
+--+--+--+--+--+--+--+
| "  Z          | Z |
+--+--+--+--+--+--+--+
| I  "  "          |
+ + +--+--+ +--+--+ +
| Z | M  | U          |
+--+--+--+--+--+--+--+
0
0
+
```

Uitvoer bij de voorbeeldinvoer

```
2 ZWAARDEN BEREIKBAAR
6 DEUREN OPENEN
===
UITGANG NIET BEREIKBAAR
===
```

Nederlands Kampioenschap Programmeren 1991

Opgave E - Kruiswoordraadsels

Je bent aangenomen door de redactie van de Baarnse Bazuin (een klein lokaal krantje) om een programma te schrijven dat hen een vervelend werkje uit handen zal nemen : het maken (niet oplossen !) van het dagelijkse kruiswoord raadsel. Het programma krijgt als invoer een lijst met omschrijvingen en woorden die aan die omschrijvingen voldoen en de vorm van het kruiswoordraadsel. Als uitvoer zal het de puzzel moeten laten zien zoals die door de lezers ingevuld moet worden en de lijst met woordomschrijvingen die daar bij hoort.

Beschrijving van de invoer

De eerste regel van de invoer bevat het aantal runs. Elke run bestaat uit twee delen. Het eerste deel bevat de omschrijvingen en de woordenlijst. Het bestaat uit een aantal omschrijvingen (minstens 1), ieder gevolgd door een aantal (minstens n) woorden die aan deze omschrijving voldoen, gevolgd door een regel waar alleen een punt op staat. Na het laatste woord behorend bij de laatste omschrijving volgt een regel waar niet n maar twee punten op staan. Iedere omschrijving en woord staat op een eigen regel en bevat alleen kleine letters, dus geen spaties of leestekens. Iedere omschrijving en woord is minstens twee en hoogstens twintig karakters lang. Er komen maximaal tien omschrijvingen voor en per omschrijving maximaal vijf woorden.

Het tweede deel beschrijft voor elke run de vorm van de puzzel. Eerst komt een regel met twee gehele getallen, K en R. K is het aantal kolommen van de puzzel, R het aantal rijen. K en R zijn minstens 1, K is hoogstens 60 en R is hoogstens 20.

Daarna komen R rijen van K karakters, punten en x-jes. Een punt betekent dat daar geen letter van een woord mag komen, een x-je betekent dat daar wel een letter van een woord moet komen. De invoer zal zo gevormd zijn dat er hoogstens 100 woorden ingevuld hoeven te worden.

De uitvoer moet voor elke run bestaan uit drie delen, het eerste deel is de ingevulde puzzel. Ook deze moet uit R rijen van K karakters bestaan (punten en kleine letters). Overal waar in de invoer-puzzel een punt staat hoort in de uitvoer-puzzel ook een punt te staan, overal waar in de invoer-puzzel een x staat moet in de uitvoer een letter van een van de woorden staan. De puzzel moet z worden ingevuld dat van links naar rechts en van boven naar beneden alleen woorden uit de bijbehorende invoer-woordenlijst voorkomen. Omschrijvingen mogen dus niet worden gebruikt, tenzij ze in de invoer ook in de woordenlijst voorkomen. Woorden mogen nul, n of meerdere keren gebruikt worden.

Het tweede deel en derde deel zijn de omschrijvingenlijst, eerst het deel met de horizontale woorden, dan het deel met de verticale woorden. Elke omschrijving moet op een eigen regel, voorafgegaan door het plaatsnummer en n spaties. Binnen een deel moeten de omschrijvingen gesorteerd worden op plaatsnummer. Die nummers kan je uit de vorm van de puzzel afleiden. Elk woordbegin krijgt een nummer, beginnend bij 1 en oplopend van boven naar beneden en op een regel van links naar rechts.

De uitvoer van de verschillende runs zijn gescheiden door een witregel.

In het algemeen zouden er bij een invoer meerdere correcte uitvoeren mogelijk kunnen zijn, of misschien geen enkele. Alle test-invoeren zullen echter zo gekozen worden dat er precies één correcte uitvoer is!

Voorbeeldinvoer

```
1
automerk
honda
porsche
mazda
.
universiteit
rul
stanford
vu
virginia
victoria
.
land
nederland
amerika
australie
.
computer
amiga
next
sparcstation
.
taal
pascal
modula
.
organisatie
acm
swep
..
20 10
.....
..x..xxxxx.....
..xxx.x..x.....
..x...x..x.....
..x...x..x.....
..xxxxxx..x.....
.....x..x.....
.....xxx.....
.x.....
xxx.....
```

Uitvoer bij de voorbeeldinvoer

```
.....
..m..amiga.....
..acm.o..m.....
..z...d..e.....
..d...u..r.....
.pascal..i.....
.....a..k.....
.....acm.....
.v.....
rul.....
horizontaal
```

2 computer
5 organisatie
6 taal
7 organisatie
9 universiteit
vertikaal
1 automeerk
3 taal
4 land
8 universiteit

Nederlands Kampioenschap Programmeren 1991

Opgave F - Signatuur

Om het opzoeken van documenten te versnellen wordt gebruik gemaakt van keywords. Ieder document heeft n of meerdere keywords. Van ieder van deze keywords kan een signatuur worden berekend. Vervolgens kunnen we deze combineren om de signatuur van het document te berekenen. Wanneer nu een gebruiker met een vraag aankomt, kunnen we ook daar de signatuur van berekenen, en deze vergelijken met die van het document. Dit kan worden gebruikt om na te gaan of het document in aanmerking komt om aan de gebruiker te laten zien.

De signatuur van een woord is de som van

$\text{ord}(c[i]) * \exp(i)$,

waarbij $c[i]$ het i -de karakter van het woord is, en $\exp(i)$ gelijk is aan 2^i tot de macht i . Alleen de eerste 16 (of minder, als het woord minder dan 16 karakters bevat) karakters van een woord worden in de signatuur betrokken. Een woord kan spaties bevatten. Leading spaces worden niet meegeteld.

De signatuur van het document is gelijk aan de logische OR van de signaturen van de keywords (de signaturen in binaire vorm).

De signatuur van de vraag wordt op dezelfde manier berekend.

De vraag matcht met het document als de signatuur van de vraag die van het document logisch impliceert, dat wil zeggen als het i -de bit in de vraag 1 is, dan is ook het i -de bit in het document 1.

Beschrijving van de invoer

De eerste regel van de invoer bevat het aantal runs. Elke run bestaat uit twee delen. In het eerste deel staan de keywords van het document, een woord per regel. Het document heeft altijd minstens 1 keyword. Dit deel wordt afgesloten met een lege regel.

Het tweede deel bestaat uit een aantal vragen, een vraag per regel (maximaal 255 karakters). De vraag heeft de vorm van een aantal (minstens n) woorden, gescheiden door komma's. Ook voor het tweede deel geldt dat het wordt afgesloten met een witregel.

Beschrijving van de uitvoer

De uitvoer bestaat uit evenveel regels als er vragen zijn. Op elke regel staat een "j" als het document met de corresponderende vraag matcht, anders een "n". De uitvoer van verschillende runs wordt gescheiden door een witregel.

Voorbeeldinvoer

paddestoel
kabouter
zwam
herfstbos

wasmachine, centrifuge
paddestoel
beukenbos, limburgse heidelandschappen

Uitvoer bij de voorbeeldinvoer

n
j
n


```
lengthe = 5.000  
lengthe = 26.155
```

Nederlands Kampioenschap Programmeren 1991

Opgave H - De Egyptische balken

Het verhaal gaat dat de bouwmeester van Cleopatra haar woede opwekte en tot de dood op de brandstapel werd veroordeeld. Hij wist echter te bedingen dat hij op een fraai geconstrueerde brandstapel zou worden verbrand, die moest worden samengesteld uit speciaal door hem geselecteerde balken. Elke balk had een aantal inkepingen. Bij goed stapelen paste elke balk precies in de inkepingen van de boven- en onderliggende balken. Dit bleek echter zø moeilijk, dat de goede man nog jaren zijn terechtstelling mocht afwachten, tot hij van ouderdom stierf (hij is uiteindelijk 138 jaar geworden !). Uw opdracht is simpel : schiet Cleopatra te hulp.

Een stapel wordt gevormd door de balken haaks op elkaar te leggen. Op een "verdieping" liggen dan twee evenwijdige balken. Op de volgende "verdieping" liggen weer twee evenwijdige balken, maar deze liggen dus loodrecht op de vorige "verdieping".

Als we een hoek van de stapel bekijken, dan kan het alleen zo zijn dat twee balken op elkaar liggen wanneer een van beide een inkeping heeft in deze hoek; f de onderste balk heeft een inkeping aan de bovenkant, f de bovenste balk heeft een inkeping aan de onderkant. Het spreekt voor zich dat balken gedraaid mogen worden, waardoor er elke balk op vier manieren kan worden neergelegd.

Voor zowel de bovenkant als de onderkant geldt dat er geen inkepingen zichtbaar mogen zijn.

Op de eerste regel van de invoer staat het aantal runs. Elke run beslaat acht regels met op elke regel de vorm van een balk. De vorm van een balk wordt beschreven door vier nullen en enen. Een n betekent een inkeping, een nul betekent geen inkeping. Het eerste cijfers geeft aan of er rechtsboven een inkeping is, het tweede cijfer geeft aan of er rechtsonder een inkeping is, het derde cijfer geeft aan of er linksboven een inkeping is en het vierde cijfer geeft tenslotte aan of er linksonder een inkeping is.

In de uitvoer moet voor elke run op een eigen regel komen te staan of er met de gegeven balken een mooie stapel te vormen is. Als dit het geval is dan moet er "stapelbaar" worden weggeschreven en anders "niet stapelbaar".

Voorbeeldinvoer

```
2
1111
1110
1101
1100
1011
1010
1001
1000
0000
0000
1111
1111
0000
```

0000
0101
0101

Uitvoer bij de voorbeeldinvoer

niet stapelbaar
stapelbaar