Nederlands Kampioenschap Programmeren 1993

Probleem A - Aircraft Conflict Detection

Om de verkeersleider te assisteren bij het verwerken van de steeds maar toenemende verkeersdrukte is er besloten om een programma te schrijven dat van te voren kan bepalen of twee vliegtuigen in de (nabije) toekomst in een conflictsituatie betrokken zullen raken (m.a.w. te dicht bij elkaar komen). Bij een conflictsituatie is de horizontale afstand tussen twee vliegtuigen minder dan 5 nautische mijlen (9260 meter).

Voor het bepalen van de positie van een vliegtuig op tijdstip t, wordt er van uitgegaan dat een vliegtuig langs een van te voren bepaalde route vliegt. Deze route wordt vastgelegd door 2 of meer way-points. Voor deze opgave is een way-point niet meer dan een naam en een (x,y)-positie.

Ter vereenvoudiging van dit conflictonderzoek mag er van worden uitgegaan dat:
- de aarde plat is,
- vliegtuigen een constante snelheid hebben,
- bochten instantaan genomen kunnen worden,
- vliegtuigen niet afwijken van hun route,
- alle vliegtuigen op dezelfde hoogte vliegen en
- vliegtuigen bij binnenkomst (aankomst op het eerste way-point) conflictvrij zijn.

Opgave

Gevraagd is een lijst met de vliegtuigparen die in een conflictsituatie raken en de bijbehorende tijdsintervallen van die conflictsituaties.

Beschrijving van de invoer

Op de eerste regel van de invoer staat een getal t dat het aantal testruns aangeeft. Hierna volgt t maal een probleemspecificatie. Een probleemspecificatie bestaat uit een lijst van way-points gevolgd door een lijst van vliegtuigen. Een lijst way-points begint op een nieuwe regel met het aantal way-points w. Hierna volgen w regels met op iedere regel een way-point. Een way-point bestaat uit een identificatie (max. 10 letters), een spatie en de positie. De positie bestaat uit een x- en y-co rdinaat gescheiden door een spatie. Zowel de x-co rdinaat als de y-co rdinaat zijn opgegeven in hele meters. De lijst vliegtuigen begint op een nieuwe regel met het aantal vliegtuigen a. Hierna volgen a regels met op iedere regel een vliegtuig. Een vliegtuig wordt ge dentificeerd door zijn callsign (max. 9 letters en/of cijfers). Na de callsign volgt een spatie, een tijdstip (in hele seconden na middernacht) waarop hij het eerste way-point zal passeren, een spatie en de snelheid v ($v > 0$ en in hele meters per seconde). Tot slot staat op dezelfde regel de route, bestaande uit het aantal way-points in de route (2 tot en met 9) en de namen van de way-points, ieder gescheiden door een spatie.

Beschrijving van de uitvoer

De uitvoer bestaat uit een lijst met daarop alle conflictsituaties. Iedere conflictsituatie komt op een

aparte regel en bevat:
- de callsigns van de twee betrokken vliegtuigen (gesorteerd op invoerfile-volgorde), gescheiden door 1 spatie, en daarachter, weer gescheiden door 1 spatie,
- het tijdsinterval gedurende welke de conflictsituatie optrad. Dit tijdsinterval staat tussen ronde haken '(' en ')' en bevat de start- en eindtijdstippen van de conflictsituatie, gescheiden door '-'.

De tijdstippen dienen te zijn gegeven in hele seconden, waarbij 0.5 naar boven afgerond dient te worden. Na iedere testrun dient een lege regel te volgen.

In de uitvoer dienen de betrokken vliegtuigen volledig op invoerfile-volgorde gesorteerd te worden. Houd er rekening mee dat twee vliegtuigen in uitzonderingsgevallen meerdere tijdsintervallen kunnen hebben waarvoor zij in conflict zijn. De conflictsituaties dienen dan in chronologische volgorde te worden gegeven, gescheiden door n spatie. Als n van beide vliegtuigen van een vliegtuigpaar het laatste way-point passeert, houdt het conflictonderzoek op. Als er op dit tijdstip sprake is van een conflictsituatie dient als eindtijd 'END' te worden ingevuld.

Voorbeeldinvoer

```
1
16
Redfa -155111 -18830
Rekken -137383 -17186
Eelde 127692 97489
Tulip -60986 7442
Flevo 57857 9307
Lilsi 47825 25759
Pampus 22956 3448
Spykerboor 6696 26302
Nyke 52590 -7896
Bergi -26690 49585
Lekko 382 -43484
Arkon 131481 -37954
Beeno -114073 102561
Nicky -39824 -126771
Sitko -124556 83437
Petik -36900 34936
5
DLH5565 0 180 5 Redfa Tulip Pampus Nyke Arkon
DLH1672 300 250 5 Rekken Flevo Spykerboor Bergi Beeno
SAS591 360 210 5 Eelde Lilsi Pampus Lekko Nicky
BAW5152 500 230 4 Sitko Petik Pampus Nyke
BAW5153 40 180 5 Sitko Petik Pampus Nyke Arkon
```

Uitvoer bij de voorbeeldinvoer

```
DLH5565 DLH1672 (730-918)
DLH5565 SAS591 (996-1052)
DLH5565 BAW5153 (908-END)
DLH1672 SAS591 (965-1002)
DLH1672 BAW5153 (879-997)
SAS591 BAW5153 (983-1026)
```

Nederlands Kampioenschap Programmeren 1993

Probleem B - Beestenmarkt

Een boer heeft last van een rekenkundige tik. Iedere keer als hij naar de markt gaat om een aantal dieren te kopen, wil hij precies net zoveel dieren kopen als het bedrag in guldens dat hij heeft meegenomen. Tevens wil hij van al zijn geld af en van ieder diersoort in ieder geval n exemplaar kopen. Hij wil alleen hele levende beesten kopen.

Opgave

Gevraagd wordt een programma dat, gegeven het bedrag dat de boer heeft meegenomen en de prijzen van iedere diersoort, de boer helpt te bepalen hoeveel van ieder soort gekocht moet worden, zodanig dat aan zijn tik is voldaan. E n oplossing is al voldoende.

Beschrijving van de invoer

Op de eerste regel staat een getal m, dat aangeeft hoe vaak de boer naar de markt gaat. Hierna volgt m maal een probleemomschrijving. Een probleemomschrijving begint op een nieuwe regel met het gehele bedrag b (0 <= b <= 100) dat de boer op zak heeft. Hierna volgt op de volgende regel een geheel getal d (1 <= d <= 10), dat het aantal soorten dieren aangeeft. Dan volgen d regels met op iedere regel een stuksprijs p (p is Real en 0 < p <= 100) van een diersoort, welke tot op de cent nauwkeurig gegeven zijn.

Beschrijving van de uitvoer

De uitvoer bestaat uit m regels met op iedere regel een antwoord. Als een oplossing gevonden was, dan dient in dezelfde volgorde als in de invoer de aantallen per diersoort gegeven te worden, gescheiden door spaties. Als er geen oplossing mogelijk was dient de tekst 'Niet mogelijk' gegeven te worden.

Voorbeeldinvoer

```
2
100
3
0.50
3
10
10
2
0.40
1.90
```

Uitvoer bij de voorbeeldinvoer

```
94 1 5
6 4
```

Nederlands Kampioenschap Programmeren 1993

Probleem C - Briks

In een vierkant schema van 5 bij 5 staan hoofdletters. Een vakje in het schema kan leeg zijn. De schikking voldoet aan de volgende voorschriften:
- Als een letter niet op de onderste regel staat, staat er een andere letter direct onder (het schema is onderhevig aan zwaartekracht, dus letters kunnen niet zweven);
- Twee aangrenzende vakken (naast of boven elkaar) bevatten verschillende letters.

Een zet in dit schema bestaat uit het verplaatsen van een willekeurige letter naar een vrije ruimte direct naast (links of rechts van) de oorspronkelijke plaats van de letter. Na een zet worden bovengenoemde voorschriften hersteld door het herhaald toepassen van de volgende regels:
- Een letter, die niet op een andere letter staat, 'valt' naar beneden tot hij wel op een andere letter rust;
- Twee of meer dezelfde letters die aan elkaar grenzen verdwijnen, maar als een 'vallende' letter even langs een gelijke letter komt, leidt dit niet tot een verdwijning.

Het doel is zodanige zetten te doen dat na enige tijd alle letters verdwenen zijn.

Opgave

Maak een programma dat gegeven een schema uitzoekt of een oplossing voor het probleem bestaat. Als er een oplossing bestaat dan dient de reeks zetten gegeven te worden die tot een oplossing leiden. Bestaat er geen oplossing dan dient de tekst 'Geen oplossing' gegeven te worden.

Beschrijving van de invoer

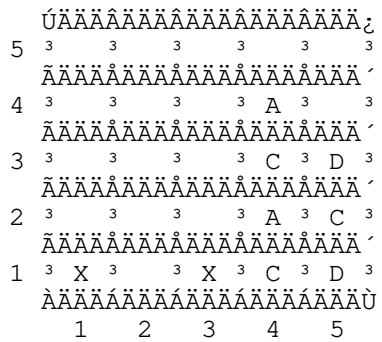
Op de eerste regel van het invoerfile staat een getal s dat het aantal schema's in het invoerfile voorstelt. Hierna volgt s maal een schema. Een schema bestaat uit 5 regels met op iedere regel 5 karakters. De eerste regel is de hoogste rij van het schema en de vijfde regel is de onderste rij van het schema. Een karakter in het invoerfile is een hoofdletter ['A'..'Z'] of een min-teken, waarbij een min-teken een lege plaats aangeeft. Een schema in het invoerfile voldoet aan bovengenoemde voorschriften.

Beschrijving van de uitvoer

De uitvoer bestaat uit s regels. Per probleem dient een regel gegenereerd te worden met een reeks zetten die tot de oplossing leiden. Als er geen oplossing mogelijk is, dus er bestaat geen reeks zetten die tot een leeg schema leiden, dan dient de tekst 'Geen oplossing' gegeven te worden. Een zet wordt weergegeven door eerst het rijnummer, direct daarna het kolomnummer, direct gevolgd door een 'R' als de letter naar rechts geschoven dient te worden of 'L' als de letter naar links geschoven dient te worden. Bijvoorbeeld '34R' betekent 'verplaats de letter in rij 3 en kolom 4 naar rechts'. Zetten dienen te zijn gescheiden door n spatie. De invoer is zo gekozen dat een oplossing altijd op regel (80 karakters) past.

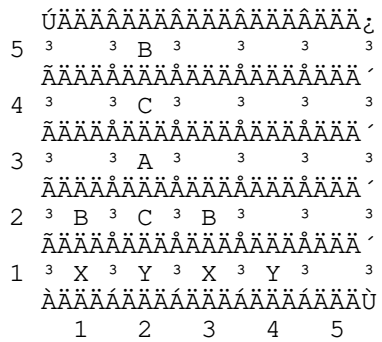
Voorbeeld

Het volgende schema



heeft onder andere als mogelijke oplossingen
44L 11R
en
24L 13L

Het volgende schema heeft geen oplossing



Voorbeeldinvoer

```
2
-----
---A-
---CD
---AC
X-XCD
-B---
-C---
-A---
BCB--
XYXY-
```

Uitvoer bij de voorbeeldinvoer

```
44L 13L
Geen oplossing
```

Nederlands Kampioenschap Programmeren 1993

Probleem D - Getalzoeker

Een rechthoekig schema is gevuld met cijfers. Uit een dergelijk schema kunnen getallen gelezen worden. Een getal ontstaat uit een opeenvolgende rij cijfers. Een dergelijke rij mag
- horizontaal, van links naar rechts en van rechts naar links,
- verticaal, van boven naar beneden en van beneden naar boven,
- diagonaal, van rechtsonder naar linksboven en terug,
- diagonaal, van linksonder naar rechtsboven en terug,

worden gelezen. Een getal mag overal beginnen en eindigen met als uitzondering dat een getal niet met een nul mag beginnen.

Men dient alle getallen die deelbaar zijn door een gegeven getal te schrappen. Een cijfer mag meer dan n maal geschrapt worden.

Opgave

Schrijf een programma dat uit elk van deze schema's de getallen schrapt die deelbaar zijn door een gegeven deler m en alle resterende cijfers als getal afdrukt.

Beschrijving van de invoer

Het invoerfile begint met het getal s dat het aantal schema's aangeeft. Hierna volgt s maal een probleemomschrijving. Een probleembeschrijving bestaat uit een regel met de deler m (1 <= m < 1ú1010) en een schemabeschrijving. Een schemabeschrijving begint met een regel met de breedte b (1 <= b < 10), een spatie en de hoogte h (1 <= h < 10). Hierna volgen h regels met b cijfers zonder tussenruimte.

Beschrijving van de uitvoer

De uitvoer bestaat uit s regels. Nadat uit een schema uit het invoerfile alle getallen zijn geschrapt die deelbaar zijn door het getal m, dienen de cijfers die niet geschrapt zijn achter elkaar te worden afgedrukt. Hierbij dient het schema van linksboven per regel naar rechtsonder te worden uitgelezen. Iedere regel dient te worden afgesloten met een punt.

Voorbeeld

```
ÚÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄ¿
³ 1 ³ 2 ³ 3 ³ 4 ³ 5 ³
ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
³ 4 ³ 2 ³ 6 ³ 8 ³ 3 ³
ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
³ 0 ³ 9 ³ 3 ³ 4 ³ 6 ³
ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
³ 4 ³ 5 ³ 3 ³ 7 ³ 9 ³
ÀÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÙ
```

In bovenstaand schema komen zo onder meer de getallen 29, 58, 493 en 4321 voor. Als gevraagd was om alle getallen deelbaar door 29 te schrappen, moeten juist deze getallen geschrapt worden en blijft het volgende schema over.

```
ÚÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄ¿
³   ³   ³   ³   ³   ³
ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
³   ³   ³ 6 ³   ³ 3 ³
ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
³ 0 ³   ³ 3 ³ 4 ³ 6 ³
ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
³ 4 ³ 5 ³   ³ 7 ³ 9 ³
ÀÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÙ
```

Achter elkaar gezet leveren deze cijfers het getal 6303464579 op.

Voorbeeldinvoer

```
2
29
5 4
12345
42683
09346
45379
2
8 4
61850266
10395739
18492849
06328252
```

Uitvoer bij de voorbeeldinvoer

```
6303464579.
.
```

Nederlands Kampioenschap Programmeren 1993

Probleem E - SimCity

Gegeven een rechthoekige stad bestaande uit m bij n vakjes. Een vakje kan zijn bebouwd met een huis (H), een fabriek (F) of een sportveld (S). Als een vakje nog niet bebouwd is, is het vrij (V). De gemeenteraad wil n van de vrije vakjes bebouwen, maar weet nog niet of het vakje met een huis, een fabriek of een sportveld bebouwd moet worden.

Uiteindelijk heeft de raad een bepaalde formule bedacht waarmee een waarde W kan worden bepaald van de hele stad. In deze formule zijn allerlei economische, sociologische en psychologische aspecten tot uitdrukking gebracht in n waarde van de stad. De hierbij gebruikte definitie van afstand is de zogenaamde Manhattan-afstand. Deze wordt bepaald door het minst benodigde aantal horizontale plus verticale stapjes om van de n naar de ander te komen.

Hoe meer fabrieken in een stad aanwezig zijn, hoe meer werkgelegenheid er bestaat. Hoe meer huizen er zijn, hoe groter de stad. De ontspanning voor de stedelingen mag niet te ver weg zijn. Voor elk huis wordt de afstand tot het dichtstbijzijnde sportveld bepaald. De grootste van deze afstanden is de ontspanningsafstand. Als de ontspanningsafstand niet kan worden bepaald, dient hiervoor de maximale afstand plus n (=m+n-1) te worden genomen. Men wil natuurlijk zo ver mogelijk bij een fabriek vandaan wonen in verband met stank en geluidsoverlast. Voor elk huis wordt de afstand tot de dichtbijzijnde fabriek bepaald. De kleinste van van deze afstanden is de welzijnsafstand. Als de welzijnsafstand niet kan worden bepaald, dient hiervoor ook de maximale afstand plus n (m+n-1) genomen te worden.

W wordt als volgt berekend:
W = aantal fabrieken + aantal huizen + aantal sportvelden
- 2 * ontspanningsafstand + 2 * welzijnsafstand

Opgave

Bepaal nu, bij een gegeven stad, of er een bebouwing van n vrij veld mogelijk is die de waarde van de stad vergroot, en zo ja, welk(e) vakje(s).

Beschrijving van de invoer

Op de eerste regel van het invoerfile staat een getal s dat het aantal steden aangeeft waarvan bepaald moet worden of de waarde vergroot kan worden. Hierna volgt s maal een stadbeschrijving. Een stadbeschrijving begint op een nieuwe regel met het getal m (1 <= m <= 20), een spatie en het getal n (1 <= n <= 20), waarbij m en n het formaat van de stad weergeven. Hierna volgen n regels met op iedere regel m veldkarakters uit de verzameling {'F','H','S','V'}.

Beschrijving van de uitvoer

De uitvoer bestaat uit s regels. Voor iedere stadsbeschrijving dienen op n regel die vrije vakjes te worden gegeven die bij bebouwing de waarde van de stad vergroten. Een vakje dient te worden

gegeven als co rdinatenpaar in de vorm '(x,y)', waarbij x (1 <= x <= m) de x-co rdinaat van het betreffende vakje en y (1 <= y <= n) de y-co rdinaat van het vrije vakje voorstellen. De co rdinaten x en y worden gerekend vanaf 'linksboven', dus het eerste veldcharacter van een stadbeschrijving uit de invoer is het punt (1,1). De co rdinatenparen dienen in volgorde van 'linksboven naar rechts' gegeven te worden. Tevens dienen de co rdinatenparen te zijn gescheiden door n spatie. De invoer zo gekozen dat de uitvoer van n test altijd op een regel van 80 karakters past.

Voorbeeld

```
     1   2   3   4   5
   ÚÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄÂÄÄÄ¿
 1 ³ H ³ V ³ F ³ S ³ V ³
   ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
 2 ³ H ³ S ³ H ³ H ³ H ³
   ÃÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄ´
 3 ³ H ³ H ³ V ³ S ³ F ³
   ÀÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÁÄÄÄÙ
```

De huidige waarde van de stad is hier 10 (2+7+3-2*2+2*1).

```
     1   2   3
   ÚÄÄÄÂÄÄÄÂÄÄÄ¿
 1 ³ F ³ V ³ S ³
   ÃÄÄÄÅÄÄÄÅÄÄÄ´
 2 ³ V ³ V ³ H ³
   ÀÄÄÄÁÄÄÄÁÄÄÄÙ
```

De huidige waarde van de stad is hier 7 (1+1+1-2*1+2*3).

Voorbeeldinvoer

```
2
5 3
HVFSV
HSHHH
HHVSF
3 2
FVS
VVH
```

Uitvoer bij de voorbeeldinvoer

```
(2,1) (5,1) (3,3)
(2,2)
```

Nederlands Kampioenschap Programmeren 1993

Probleem F - N-Ster

Een regelmatige n-hoek, met n >= 3, heeft (n(n-2))/2 diagonalen: lijnen die twee hoekpunten verbinden die niet door een zijde verbonden zijn. Voor n >= 5 zijn er n van deze diagonalen die twee hoekpunten verbinden waartussen zich precies n ander hoekpunt bevindt. Deze laatste diagonalen vormen de zijden van een regelmatige n-ster (zie figuur).

De snijpunten van de zijden van een n-ster noemen we zijn hoekpunten. Een n-ster heeft 2n van zulke hoekpunten. De helft daarvan valt samen met de hoekpunten van de oorspronkelijke n-hoek. Deze hoekpunten worden buitenhoekpunten genoemd. De andere hoekpunten van de n-ster liggen in het inwendige van de oorspronkelijke n-hoek. Deze hoekpunten worden binnenhoekpunten genoemd.

Nederlands Kampioenschap Programmeren 1993

Probleem G - Paardesprong

Tussen de opnames van "Ook dat nog!" door heeft Hans B hm nog tijd voor een spelletje schaak. Omdat niemand van het "Ook dat nog!"-team nog tegen Hans wil spelen (Gregor: "Hans wint alles..."), heeft hij een andere bezigheid gevonden.

Het spelletje wordt gespeeld met een rechthoekig bord van willekeurige afmetingen en een paard. Iemand van het "Ook dat nog!"-team noemt de afmetingen van het bord en de plaats waar het paard als eerste wordt neergezet. Hans belooft dat hij na een tijdje puzzelen zal zeggen of hij met behulp van het paard en de paardesprong alle vakjes van het bord precies n keer kan betreden. De beginplaats is al betreden door het paard neer te zetten en het paard hoeft daar niet te eindigen.

Bij kleine borden kon men Hans' uitspraken nog wel controleren, maar bij grotere borden werd dat toch wel problematisch. Daarom roept het "Ook dat nog!"-team jouw hulp in om een programma te schrijven dat de uitspraken van Hans kan controleren.

Paardesprong

Een paardesprong is bijvoorbeeld twee vooruit, n naar links. Hieronder is een schema gegeven met de velden die een paard kan bereiken in n zet. Het paard staat dan op de positie die met een 'P' een gemerkt en de velden die met een 'x' zijn gemerkt kunnen in n zet bereikt worden.

```
ÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅ
  ³     ³  x  ³     ³  x  ³     ³     ³
ÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅ
  ³  x  ³     ³     ³     ³  x  ³     ³
ÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅ
  ³     ³     ³  P  ³     ³     ³     ³
ÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅ
  ³  x  ³     ³     ³     ³  x  ³     ³
ÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅ
  ³     ³  x  ³     ³  x  ³     ³     ³
ÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅÄÄÄÅ
```

Opgave

Schrijf een programma dat controleert of een bord van m (1 <= m <= 10) bij n (1 <= n <= 10) vanaf beginplaats (a,b) (1 <= a <= m en 1 <= b <= n) helemaal bewandeld kan worden met een paard.

Beschrijving van de invoer

De eerste regel van het invoerfile bevat het getal p dat aangeeft hoeveel probleemspecificaties er volgen. Dan volgen er p probleemspecificaties.

Een probleemspecificatie begint op een nieuwe regel en bestaat uit twee regels. De eerste regel bevat de afmeting van het bord en bestaat uit het getal m, een spatie en het getal n, waarbij m het

aantal kolommen aangeeft en n het aantal rijen. De tweede regel bevat de plaats waar het paard begint en bestaat uit het getal a, een spatie en het getal b, waarbij a het kolomnummer aangeeft en b het rijnummer.

Beschrijving van de uitvoer

Het uitvoerfile dient te bestaan uit p regels. Per probleemspecificatie uit het invoerfile dient de tekst ’ja’ gegeven te worden als het mogelijk is het gehele bord te doorlopen en de tekst ’nee’ als dit niet mogelijk is.

Voorbeeldinvoer

```
3
3 3
2 2
8 8
1 1
3 2
1 1
```

Uitvoer bij de voorbeeldinvoer

```
nee
ja
nee
```

Nederlands Kampioenschap Programmeren 1993

Probleem H - De toffe tegelzetter

Gegeven is een ruimte met een vloer van m bij n meter. In deze ruimte zijn obstakels aanwezig zoals pilaren en stenen bloembakken. Men wil deze vloer opnieuw betegelen. De tegels die men wil gebruiken hebben n bepaalde vorm. Deze vorm kan worden omschreven als een rechthoek, waar stukken uitgehaald zijn.

Opgave

Gevraagd is of het mogelijk is om de vloer te betegelen zonder tegels te versnijden. De tegels mogen worden gedraaid en gekeerd (een tegel heeft twee 'bovenkanten').

Beschrijving van de invoer

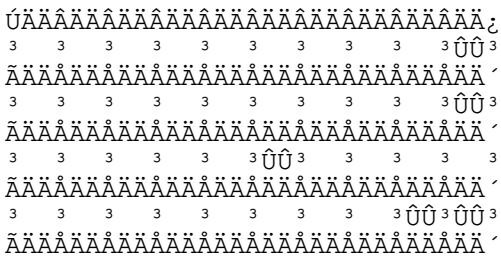
De eerste regel van de invoer bevat het getal p dat het aantal problemen voorstelt dat opgelost moet worden. Hierna volgt p maal een probleembeschrijving. Een probleembeschrijving bestaat uit een tegelbeschrijving en een vloerbeschrijving. Een tegelbeschrijving begint op een nieuwe regel met het getal a (1 <= a <= 5), een spatie en het getal b (1 <= b <= 5), welke de omhullende rechthoek van de tegel voorstellen. Hierna volgen b regels met op iedere regel a enen en nullen. De vorm van de tegel wordt bepaald door de enen. De tegel is samenhangend, d.w.z. de tegel kan niet uit afzonderlijke componenten bestaan. Tevens bevatten de tegels geen gaten. Een vloerbeschrijving begint op een nieuwe regel met het getal m (0 <= m <= 20), een spatie en het getal n (0 <= n <= 20). Hierna volgen n regels met m enen en nullen. Een nul betekent dat het veld vrij is. Een n betekent dat het betreffende veld bezet is door een pilaar of een ander obstakel en dus niet betegeld hoeft te worden.

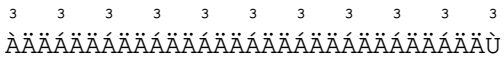
Beschrijving van de uitvoer

De uitvoer dient te bestaan uit v regels. Per vloer uit de invoer dient de tekst 'ja' gegenereerd te worden als er een oplossing mogelijk was en de tekst 'nee' als de vloer niet met die tegel betegeld kon worden.

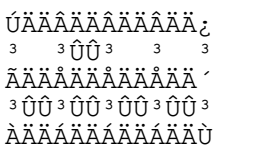
Voorbeeld

Vloer (m=10,n=5)

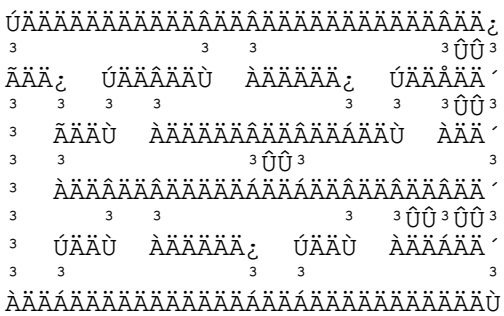




Tegel (a=4,b=2)



Deze vloer kan als volgt betegeld worden:



Het antwoord is dus ja.

Voorbeeldinvoer

```
2
4 2
0100
1111
10 5
0000000001
0000000001
0000010000
0000000011
0000000000
4 2
1111
1110
20 3
00000000000000000000
00000000000000000000
00000000000000000000
```

Uitvoer bij de voorbeeldinvoer

```
ja
nee
```