

Voorrunde Programmeren 1993

Probleem A - AWARI

Spelregels

Awari is een Afrikaans denkspel voor twee personen. Het wordt gespeeld op een houten bord waarin twee rijen van 6 kuilen, de zgn. speelkuilen, zijn aangebracht. Elk van de spelers beheerst de rij aan zijn kant van het bord. De kuilen van Zuid, de eerste speler, worden van links naar rechts A t/m F genoemd (vanuit Zuid gezien). De kuilen van Noord, de tweede speler, worden voor hem gezien van links naar rechts a t/m f genoemd. Aan de rechterkant van de rij kuilen van een speler is een extra kuil gemaakt, waarin de door die speler geslagen stenen worden opgeslagen. Het spel begint met in elk van de 12 speelkuilen 4 stenen.

De speler die aan zet is, selecteert een niet-lege kuil X uit zijn rij. Beginnend met de rechterbuurkuil van X zaait hij alle stenen uit een kuil n voor n in tegenwijzerrichting over het bord, daarbij de tw extra kuilen overslaand. Als X voldoende stenen bevat (12 of meer) dan wordt bij het zaaien kuil X zelf overgeslagen. Na de zet zal X dus altijd leeg zijn. Aan het eind van de zet worden geslagen stenen verwijderd. Stenen zijn geslagen als de laatst gezaaide steen in een vijandelijke kuil terecht komt, die daardoor 2 of 3 stenen bevat. Als dit zo is, dan wordt ook naar de kuil gekeken waar de voorlaatste steen in terecht kwam. Opnieuw worden de stenen geslagen als het er 2 of 3 zijn en het een vijandelijke kuil betreft. Dit slaan gaat door, totdat er een ander aantal dan 2 of 3 stenen wordt aangetroffen, of het einde van de vijandelijke rij wordt bereikt.

Het doel van Awari is om meer stenen dan je tegenstander te slaan. De partij eindigt dan ook zodra een van de spelers 25 of meer stenen heeft geslagen. Er zijn nog twee andere condities, waaronder de partij wordt beïndigd:

1. Als een speler niet meer kan zetten (d.w.z., al zijn kuilen zijn leeg terwijl hij aan zet is). De tegenstander krijgt dan alle resterende stenen op het bord.
2. Dezelfde positie (dezelfde toestand over het hele bord gezien) wordt voor de derde keer bereikt met dezelfde speler aan zet. De resterende stenen op het bord worden dan gelijkelijk verdeeld. In alle gevallen is na afloop de winnaar degene met de meeste stenen. Bij een stand van 24-24 is het remise.

Een laatste, wat kunstmatige, regel bestaat in Awari. Het is verboden een zet te doen zodanig dat de tegenstander daarna geen zetten heeft, tenzij je niet anders kunt. Dus, wanneer er keuze is en er een zet bestaat die ervoor zorgt dat de tegenstander vervolgens stenen in zijn kuilen heeft, dan moet zo'n zet worden uitgevoerd. Als er alleen maar zetten zijn die de tegenstander zonder stenen laten, dan heeft de tegenstander pech gehad.

Notatie zetten

In de korte notatie wordt een zet beschreven door uitsluitend de naam van de kuil. In de lange notatie wordt eerst de naam van de kuil gegeven, gevolgd door het aantal stenen dat wordt uitgezaaid uit de kuil. Indien er stenen worden geslagen, volgt er een 'x' en het aantal geslagen stenen.

Een voorbeeld positie is:

Noord
f e d c b a

(geslagen Noord) 7	0	0	1	3	1	1	5	(geslagen Zuid)	
	1	0	4	19	4	2			
	A	B	C	D	E	F			
	Zuid (aan zet)								

Zuid heeft als legale zetten A,C,D,E en F (korte notatie), oftewel A1, C4x2, D19x7, E4 en F2x4 (lange notatie).

Opgave

Na afloop van een Awari toernooi wil de toernooileiding een bulletin uitgeven met alle partijen erin, in lange notatie. Nu hebben sommige deelnemers echter in korte notatie genoteerd. Daarom moet een programma geschreven worden, dat een partij in korte notatie inleest en in lange notatie afdruckt.

Beschrijving van de invoer

Het getal r op de eerste regel van het invoerfile geeft aan hoeveel spelletjes vertaald dienen te worden. Dan volgen de zetten van r spelletjes in de korte notatie. De zetten kunnen op meer regels staan, waarbij iedere regel een onbekend aantal letters bevat. Na de laatste zet van een spelletje volgt een punt. Een nieuw spel begint op een nieuwe regel.

Beschrijving van de uitvoer

Bij de te produceren uitvoer moeten de zetten genummerd worden. Een zet bestaat uit een beurt van Zuid en een beurt van Noord. Elke nieuwe zet moet op een nieuwe regel beginnen. De nummering van een zet bestaat uit het zetnummer gevolgd door een punt. Achter het zetnummer komt een spatie, gevolgd door de zet van Zuid en Noord met dit nummer. De zetten van Zuid en Noord moeten worden gescheiden door een komma die gevolgd wordt door een spatie. Na een spelletje volgt een lege regel.

Voorbeeldinvoer

```
1
FeDaFaBaEfAdCcDfFeEaCfBaAbBaDcFbEbCeCfCc
AdBaFaAfAeAdCeBbDaEaFcCaDdEbFfBaCcDeAdEb
BcCeDdEeFfCbDcEaBdCbDeAcBdCfDeAfEaFaBbCc
DdAeBfA.
```

Uitvoer bij de voorbeeldinvoer

1. F4, e4
2. D4, a6
3. F1, a1
4. B5, a1
5. E6x2, f6x3
6. A7, d7x3
7. C9x4, c8x2
8. D6x3, f1x3
9. F3x2, e1
10. E1, a5
11. C1, f2
12. B5, a1

13. A1, b16x2
14. B2, a1
15. D4x2, c3
16. F4, b1
17. E3, b1
18. C3, e4
19. C1, f4x3
20. C1, c3
21. A2, d6x4
22. B4, a3
23. F3x4, a1
24. A1, f2x2
25. A1, e2
26. A1, d1
27. C1, e1
28. B2, b1
29. D3, a1
30. E3x2, a1
31. F2x2, c1
32. C1, a1
33. D1, d1
34. E1, b1
35. F1, f2
36. B1, a1
37. C1, c1
38. D1, e1
39. A1, d1
40. E1, b1
41. B1, c1
42. C1, e1
43. D1, d1
44. E1, e1
45. F2, f3
46. C1, b1
47. D1, c1
48. E1, a1
49. B1, d1
50. C1, b1
51. D1, e1
52. A1, c1
53. B1, d1
54. C1, f1
55. D1, e1
56. A1, f1
57. E2, a1
58. F2x2, a1
59. B1, b1
60. C1, c1
61. D1, d1
62. A1, e1
63. B1, f1
64. A1

Voorrunde Programmeren 1993

Probleem B - Boemboem

Het spel Boemboem wordt gespeeld op een rechthoekig speelveld van M bij N vakjes. Sommige vakjes zijn ontoegankelijk. Deze worden niet tot het speelveld gerekend. De (toegankelijke) vakjes, die direct grenzen aan een gegeven (toegankelijk) vakje worden de burens van dat vakje genoemd. Als er geen ontoegankelijke vakjes zijn, heeft een hoekvakje dus twee, een vakje langs de rand drie en een vakje in het midden vier burens. De ligging van de ontoegankelijke vakjes is zodanig dat ieder toegankelijk vakje tenminste n buur heeft. Tevens delen de ontoegankelijke vakjes het veld nooit in twee of meer aparte delen.

Het spel wordt gespeeld door twee spelers. Speler n begint en speelt met witte stenen. Speler twee speelt met zwarte stenen. De spelers mogen om de beurt een zet doen. Een zet bestaat uit het plaatsen van een steen in een (toegankelijk) vakje. Een vakje kan meer dan n steen bevatten. Een speler mag alleen een steen plaatsen in een vakje dat al stenen van zijn kleur bevat f in een vakje dat leeg is.

Als een vakje evenveel of meer stenen bevat als het aantal burens van dat vakje, dan ontstaat een explosie. Een explosie heeft tot gevolg dat iedere buur n steen uit het exploderende vakje krijgt. Al er in een buurvakje al stenen aanwezig waren van de andere speler, veranderen de aanwezige stenen in de kleur van de steen uit het exploderende vakje. Hierdoor kan een kettingreactie ontstaan. Het spel is voorbij zodra er minimaal twee zetten zijn gedaan en alle stenen in het veld dezelfde kleur hebben. De verliezer heeft dan geen enkele steen van zijn kleur meer in het veld.

Gevraagd wordt een programma dat, gegeven een aantal zetten van beide spelers, bepaalt welke speler de winnaar is. In het geval dat er een winnaar is, dient de kleur van de winnende speler ('Wit' of 'Zwart') gegeven te worden. Is er nog geen winnaar, dan dient de tekst 'Nog onbeslist' gegeven te worden.

Beschrijving van de invoer

Het eerste getal in de invoer geeft het aantal spelletjes s aan. Hierna volgt s maal een spelbeschrijving. Een spelbeschrijving is gegeven door een veldbeschrijving gevolgd door de zetten. De zetten staan in volgorde van spelverloop.

De veldbeschrijving begint op een nieuwe regel met de getallen M ($2 \leq M \leq 50$) en N ($2 \leq N \leq 50$), welke de grootte van het veld weergeven. Dan volgt op de volgende regel een getal v dat het aantal niet-toegankelijke velden aangeeft, waarna v regels volgen met ieder een niet-toegankelijk veld.

De beschrijving van de zetten begint op een nieuwe regel met het aantal zetten z . Hierna volgen z regels met ieder een zet. Mocht een winnaar zijn bepaald voordat alle zetten zijn gedaan, dan dienen de resterende zetten te worden overgeslagen.

Een ontoegankelijk vakje en een zet worden beide aangegeven als een (x,y) -paar. Een (x,y) -paar bestaat uit een getal x ($1 \leq x \leq M$), een spatie en een getal y ($1 < y \leq N$).

Beschrijving van de uitvoer

Per spel uit de invoer dient een regel te worden gegenereerd met de tekst 'Wit', 'Zwart' of 'Nog onbeslist', afhankelijk van het resultaat van de het spel.

Voorbeeldinvoer

```
4
4 3
1
4 3
9
1 1
4 1
1 1
4 1
1 1
4 1
1 1
4 1
2 1
2 2
0
4
1 1
2 2
2 1
2 2
5 5
2
2 2
4 4
2
1 1
5 1
3 3
1
3 3
10
1 1
1 2
1 1
3 2
1 2
3 2
1 2
3 2
1 1
3 2
```

Uitvoer bij de voorbeeldinvoer

```
Wit
Zwart
Nog onbeslist
Wit
```

Voorronde Programmeren 1993

Probleem C - Dubbele-Letters-Puzzel

Voor deze puzzel worden woorden gebruikt die geschreven worden met een alfabet dat alleen bestaat uit de hoofdletters A t/m E. Deze woorden mogen worden herschreven volgens de volgende 5 herschrijfgeregels:

1. xAAy -> xEy | xBy
2. xBBy -> xAy | xCy
3. xCCy -> xBy | xDy
4. xDDy -> xCy | xEy
5. xEEy -> xDy | xAy

M.a.w. ieder voorkomen van dubbele letters (b.v. AA) mag worden vervangen door de cyclische opvolger (bij AA: B) of cyclische voorganger (bij AA: E) van de letter die dubbel voorkwam. De letters x en y in de herschrijfgeregels staan voor de woorden die voor respectievelijk na de te herschrijven dubbele letter staan. Woorden mogen leeg zijn, dat wil zeggen uit 0 letters bestaan.

Schrijf een programma dat uitzoekt tot welke woorden van n letter het gegeven woord, met behulp van de vijf genoemde herschrijfgeregels, kan worden herleid.

Beschrijving van de invoer

De eerste regel van de invoer geeft het aantal woorden aan dat getest dient te worden. Hierna volgt op iedere regel een woord, dat alleen de hoofdletters A t/m E bevat. Een woord bevat niet meer dan 20 hoofdletters.

Beschrijving van de uitvoer

Per woord uit de invoer dienen de letters te worden opgesomd, waartoe het woord kan worden herleid. De letters dienen in alfabetische volgorde te staan, alle gescheiden door n spatie. Als het woord niet herleidbaar blijkt, dient de tekst 'Niet herleidbaar' te worden gegeven.

Voorbeeldinvoer

```
3
AABDCBBDCAA
AADBAA
BBCAA
```

Uitvoer bij de voorbeeldinvoer

```
A B C E
Niet herleidbaar
A C
```


De tweede voorbeeldtest is gegeven door:

ÚÄÄÄÄÄÄÄÄÄÄÄ;ÚÄÄÄÄÄÄÄÄÄÄÄ;
3 00000000 33 00 3
3 00000000 33 00000000 3
3 00000000 33 00 3
ÄÄÄÄÄÄÄÄÄÄÄÄÜÄÄÄÄÄÄÄÄÄÄÄÜ
Originele foto's

ÚÄÄÄÄÄÄÄÄÄÄÄ;ÚÄÄÄÄÄÄÄÄÄÄÄ;
3 00 33 3
3 0000 33 3
3 0000 33 3
ÄÄÄÄÄÄÄÄÄÄÄÄÜÄÄÄÄÄÄÄÄÄÄÄÜ
Verstoorde opnamen (foto's of negatieven)

Voorbeeldinvoer

2
5
2
10001
01010
00100
01010
10001
00110
01001
10001
10010
01100
3
10101
01010
10100
00000
10001
11100
10110
01010
01111
10011
00110
00001
00001
11010
01000
3
2
111
111
111
010
111
010
2
010
011
110
000
000
000

Uitvoer bij de voorbeeldinvoer

1, positief
2, negatief
2, positief

2, positief
1, negatief

Voorrunde Programmeren 1993

Probleem E - Propagatietijd in circuits

Bij het ontwerpen van chips moet men zich met veel zaken bezig houden. Allerlei factoren limiteren de grootte en snelheid van een chip. Gevraagd wordt om een hulp- programma te schrijven dat de snelheid van een chip bepaalt.

Elke component op een chip, of het nu een transistor of een verbinding tussen onderdelen is, geeft een bepaalde vertraging: de propagatietijd. De componenten op een chip vormen een netwerkstructuur. Als componenten sequentieel geschakeld zijn, is de propagatietijd de som van de propagatietijden van de componenten. Bestaan er verschillende paden door een chip, dan is het langzaamste pad gelijk aan de propagatietijd van de hele chip.

De meeste producten, waarmee men chips kan ontwerpen, staan het toe om gemoduleerd te ontwerpen. Dit houdt in dat men standaard modules kan bouwen van basisstenen, zoals transistoren en verbindingen. Deze modules kunnen weer gebruikt worden als bouwstenen in latere modules, enz. Een module kan worden beschouwd als een 'black box' met een bepaalde propagatietijd. Er wordt veel van modules gebruik gemaakt.

Een complicatie bij deze opgave is dat een uitgangssignaal van een systeem weer alsingangssignaal van hetzelfde systeem gebruikt kan worden. U dient dit te detecteren en zodra dit voorkomt slechts een zin te produceren, die luidt 'Recursief circuit'.

Beschrijving van de invoer

Op de eerste regel van het invoerfile staat een getal dat het aantal testen aangeeft. Per test volgt op een nieuwe regel het aantal modules dat gedefinieerd is. De modules zijn gedefinieerd volgens de volgende methode: eerst komt de naam (aaneengesloten alfanumerieke rij beginnend met een letter en maximale lengte 10) dan een spatie gevolgd door 3 getallen die achtereenvolgens voorstellen: het aantal ingangcontactpunten i ($0 \leq i \leq 8$), het aantal uitgangcontactpunten u ($0 \leq u \leq 8$) en het aantal deelelementen d (submodules of basisstenen) ($0 \leq d \leq 20$) van de module. De contactpunten zijn genummerd van 1 tot i resp. u .

Vervolgens worden de deelelementen opgesomd. De volgorde van opsomming is bepalend voor het referentienummer van het deelelement. Het eerste deelelement heeft referentienummer 1, het tweede deelelement nummer 2, enz. Een deelelement is op een regel gedefinieerd door eerst de naam van de module, dan de verwijzingen van de ingangsignalen en vervolgens de verwijzingen van uitgangsignalen naar andere deelelementen. De verwijzingen zijn gescheiden door twee spaties.

Een verwijzing bestaat uit 2 getallen gescheiden door een spatie. Het eerste getal is het referentienummer van de module of het basiselement waarmee contact wordt gemaakt. Het tweede getal geeft het nummer van het contactpunt van de betreffende ingang of uitgang aan. Indien het eerste getal van zo'n verwijzing -1 is, dan duidt dit op een contactpunt buiten de module. Voor ieder contactpunt geldt, dat deze slechts met een andere component verbonden kan zijn. Er is bekend dat de modules net geordend zijn, dat wil zeggen dat er nooit naar een module wordt gerefereerd, dat pas later wordt gedefinieerd. De laatste module van een test is tevens het circuit, waarvan de propagatietijd bepaald moet worden.

Voorronde Programmeren 1993

Probleem F - Tjoeke-tjoeke tuut-t&t

De toekomstige student zal het moeilijk hebben: geen OV-jaarkaart, weinig geld en een tempo-beurs. In veel gevallen zal er bijverdiend moeten worden om toch nog te kunnen stappen, af en toe een CD-tje te kunnen kopen en eens uit eten te kunnen. In deze nabije toekomst ontstaat op een gegeven moment de volgende situatie.

Een student begint een n-manskoeriersbedrijfje. De NS heeft intussen zoveel extra zitplaatsen weten te creëren, dat zelfs in de spits genoeg ruimte is om iedereen plaats te bieden. Met de tempobeurs in het achterhoofd en het idee dat de reistijd toch nog nuttig besteed kan worden met studeren, besluit de student gebruik te maken van de trein als vervoermiddel. In korte tijd groeit het bedrijfje betrekkelijk snel. Na een paar jaar beschikt de student over een vaste klantenkring. Hij wil niet meer uitbreiden, omdat hij in zijn eentje deze situatie nog precies onder controle kan houden. Er is een stabiele situatie ontstaan.

Om de kosten te beperken wil hij gebruik maken van trajectkaarten (met deze kaart kan men gedurende een bepaalde periode onbeperkt op een bepaald traject reizen). Hij wil alleen de huidige klantenkring bedienen. Verder meent hij over zee n van tijd te beschikken en zijn enige zorg is om de reiskosten te minimaliseren.

Gevraagd wordt om gegeven de woonplaatsen van de vaste klanten, het bestaande netwerk tussen deze plaatsen van de NS en de kostprijs van de trajectkaarten, een programma te schrijven, dat de voor hem meest gunstige verzameling trajectkaarten bepaalt.

Beschrijving van de invoer

Op de eerste regel van het invoerfile staat een getal t dat het aantal testen aangeeft. Hierna volgt t maal:

- een regel met een getal n ($1 \leq n \leq 50$), waarbij n het aantal steden aangeeft;
- een regel met een getal m ($0 \leq m \leq n(n-1)/2$), waarbij m het aantal verbindingen tussen de steden aangeeft;
- een lijst van m regels met op iedere regel drie getallen. De eerste twee getallen s_1 en s_2 ($1 \leq s_i \leq n$) geven de steden aan waartussen de verbinding bestaat. Het laatste getal betreft de prijs p ($p > 0$) van de trajectkaart (op twee decimalen nauwkeurig). De getallen zijn gescheiden door een spatie.

De invoer is zo gekozen dat het altijd mogelijk is om alle steden met elkaar te verbinden.

Beschrijving van de uitvoer

Per test dient het totale bedrag gegeven te worden dat de student dient te betalen voor de minimale verzameling trajectkaarten, waarmee alle steden bereikbaar zijn. Ieder bedrag dient op een aparte regel te staan en verder dient het bedrag altijd op twee decimalen nauwkeurig te worden afgedrukt. Het decimaalteken is een punt.

Voorbeeldinvoer

2
3
3
1 3 30.35
1 2 8.45
2 3 22.95
5
5
4 5 40.45
1 3 22.35
4 2 22.45
1 5 39.50
4 3 30.00

Uitvoer bij de voorbeeldinvoer

31.40
114.30

Voorrunde Programmeren 1993

Probleem G - Voorzienige Liften

Ter illustratie een fragment uit "The Hitch Hiker's Guide to the Galaxy, A Trilogy in Four Parts" van Douglas Adams:

"Modern elevators are strange and complex entities. The ancient electric winch and 'maximum-capacity-eight-persons' jobs bear as much relation to a Sirius Cybernetics Corporation Happy Vertical People Transporter as a packet of mixed nuts does to the entire west wing of the Sirian State Mental Hospital.

This is because they operate on the curious principle of 'defocused temporal perception'. In other words they have the capacity to see dimly into the immediate future, which enables the elevator to be on the right floor to pick you up before you knew you wanted it, thus eliminating all the tedious chatting, relaxing, and making friends that people were previously forced to do whilst waiting for elevators.

Not unnaturally, many elevators imbued with intelligence and recognition became terribly frustrated with the mindless business of going up and down, up and down, experimented briefly with the notion of going sideways, as a sort of existential protest, demanded participation in the decision-making process and finally took to squatting in basements sulking.

An impoverished hitch hiker visiting any planets in the Sirius star system these days can pick up easy money working as a councillor for neurotic elevators."

Men is nog niet helemaal tevreden over het algoritme dat bepaalt in welke volgorde de lift langs de verschillende verdiepingen moet bewegen. Ondanks het vermogen beperkt in de toekomst te kijken, blijven de resultaten ver achter bij de verwachtingen.

Gevraagd wordt een geheel nieuw algoritme te ontwerpen, dat gegeven de informatie over toekomstige 'klanten', de meest efficiënte weg bepaalt. Waarbij meest efficiënt betekent dat de gemiddelde reistijd (wachtijd+transporttijd) over alle liftgebruikers en het tijdstip, waarop de lift de laatste personen heeft afgeleverd, geminimaliseerd moeten worden.

Het gebouw waar deze opgave betrekking op heeft, heeft slechts n lift. Men kan op iedere verdieping in- en uitstappen. De hoogste verdieping is verdieping 7 en de laagste is verdieping 0. Er kunnen maximaal 10 personen in de lift. Het duurt gemiddeld 10 seconden om op volle snelheid een verdieping te passeren.

Als een lift vertrekt van een verdieping, geeft dit een vertraging. Mensen moeten in- en uitstappen en nadat de deuren zijn gesloten, duurt het even voor de lift op gang is. Hetzelfde gebeurt in omgekeerde volgorde als een lift stopt. Als een lift, zodra deze net op volle snelheid is gekomen, direct weer afremt, is deze precies n verdieping hoger of lager. Deze verplaatsing van n verdieping; inclusief de vertraging bij in- en uitstappen duurt in totaal 20 seconden. Het duurt bijvoorbeeld 20 seconden om van de eerste naar de tweede verdieping te gaan en het duurt 30 seconden om van de eerste naar de derde verdieping te gaan. Als beginvoorwaarde geldt dat op tijdstip 0 de lift op verdieping 0 staat.

Beschrijving van de invoer

Op de eerste regel van de invoer staat het aantal testruns r . Hierna volgt r maal informatie over de directe toekomst. Informatie over de directe toekomst bestaat uit een regel met het aantal informatieregels i , waarna i regels volgen in strikt stijgende chronologische volgorde. Per regel is eerst het tijdstip t van aankomst ($t \geq 0$) gegeven, gevolgd door een spatie, het aantal personen p ($0 < p \leq 20$), een spatie, de verdieping van aankomst, een spatie en de verdieping van bestemming. Alle tijden zijn gegeven in hele seconden. De invoer is zo gekozen dat de lift in maximaal vier uur tijd alle verzoeken kan behandelen.

Beschrijving van de uitvoer

Uw uitvoer dient te bestaan uit een regel met de gemiddelde reistijd (wachtijd + transporttijd) per persoon, gevolgd door een spatie en het tijdstip, waarop de laatste passagier op zijn/haar bestemming is afgeleverd. De tijden in de uitvoer dienen in hele seconden (.5 afronden naar boven) gegeven te worden.

Voorbeeldinvoer

```
2
3
60 8 1 2
85 3 2 0
110 3 1 0
3
20 5 3 7
80 1 6 0
140 1 3 0
```

Uitvoer bij de voorbeeldinvoer

```
25 130
73 190
```


Voorronde Programmeren 1993

Probleem H - Vreemde Verjaardag

Magmed Ali, abt van de Magmezen, verzorgt altijd de verjaardagen van de priesters. Al weken voor een verjaardag begint de abt aan het intensieve rekenwerk om de nieuwe leeftijd van de priester via een rekenkundige expressie uit te drukken. Abt Magmed neemt dan alle leeftijden in beschouwing van de overige priesters en probeert hiermee zo'n expressie te maken dat de uitkomst van de expressie gelijk is aan de leeftijd van de binnenkort jarige priester. Iedere leeftijd mag slechts n maal worden gebruikt. Binnen zo'n expressie mogen alleen de operaties delen, vermenigvuldigen, optellen en aftrekken gebruikt worden. Delen is alleen toegestaan als het quotiënt een geheel getal is. De abt gebruikt ook haakjes. Helaas wordt Magmed zelf ook een dagje ouder en hij kan niet zo snel meer rekenen. Gevraagd wordt een programma dat aangeeft of er een oplossing bestaat en als deze bestaat, de expressie geeft.

Beschrijving van de invoer

Op de eerste regel van de invoer staat een getal dat het aantal testen t aangeeft, waarna t testen volgen. Iedere test begint met een regel met de leeftijd van de jarige priester, daarna op de volgende regel een getal p ($0 \leq p \leq 7$), dat het aantal overige priesters aangeeft. Hierna volgen p regels met op iedere regel de leeftijd l ($15 \leq l \leq 100$) van een priester.

Beschrijving van de uitvoer

Als er geen rekenkundige expressie mogelijk is, dient de regel met de tekst 'Niet mogelijk' gegeven te worden. Is er een oplossing gevonden, dan dient de gevonden expressie gegeven te worden. Een expressie wordt gedefinieerd door:

$:: \mid ' (, , , ,) '$

met

$:: ' + ' \mid ' - ' \mid ' * ' \mid ' / '$

Met andere woorden een expressie is gedefinieerd door f een getal f een rij beginnend met een haakje openen, direct gevolgd door een expressie, direct gevolgd door een operatie, direct gevolgd door een expressie en is afgesloten met een haakje sluiten. Hierbij is een operatie gedefinieerd als f een plusteken f een minteken f een maalteken f een deelstreep.

De getallen in de expressie zijn gehele getallen, waarbij ieder getal de leeftijd is van n van de overige (niet-jarige) priesters.

Voorbeeldinvoer

3
46
4
54
27
35

18
54
3
30
53
46
40
3
30
50
40

Uitvoer bij de voorbeeldinvoer

((54/18)*27)-35
Niet mogelijk
(30+50)-40