



Opgaven NKP 2001

6 oktober 2001
TU Delft

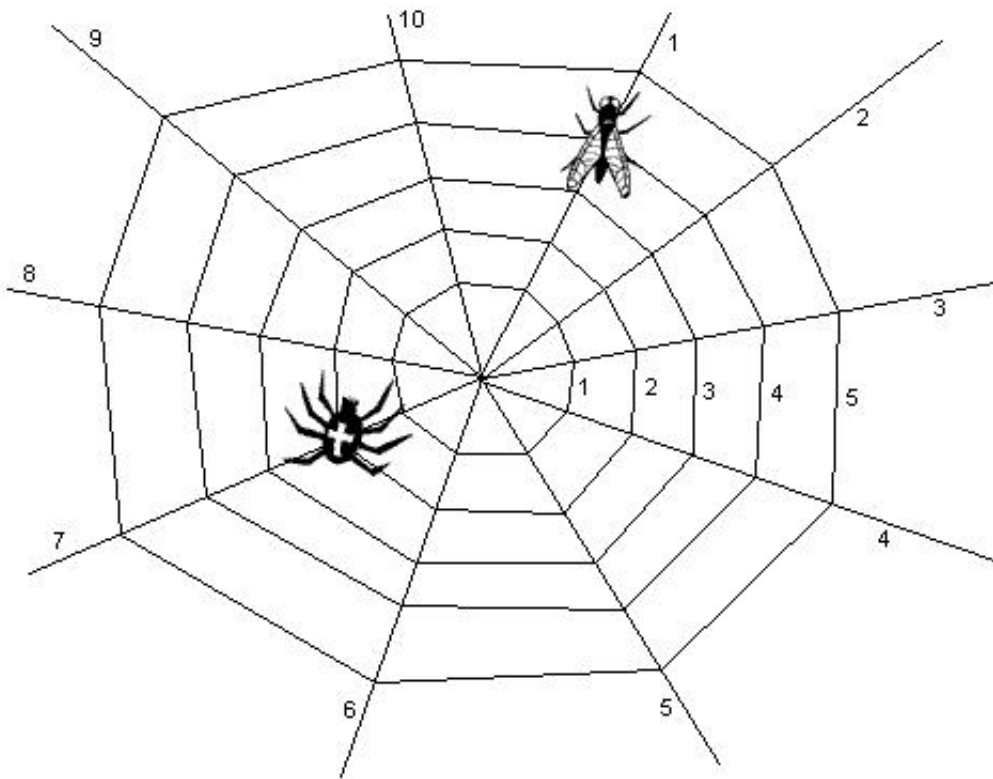
Inhoudsopgave

A De spin en zijn prooi	1
B Palindromië	3
C Boeken – weet je nog?	5
D Schrikkeljaren	7
E Binaire Codes	9
F Verkiezingen	13
G De Fusiegolf	15
H De stranden van Polygonsië	17

A De spin en zijn prooi

Zoals jullie allemaal weten, vangen spinnen hun vliegen niet met uitvouwbare, kubusvormige kamers, maar gewoon met spinnenwebben. De huisspin in studentenhuis 'Het Spinnehol' heeft zoveel te eten dat het vliegenvangen hem een beetje gaat vervelen. Ook het van te voren berekenen van de route die hem de minste stappen kost om bij de vlieg te komen vindt hij niet leuk meer. Hij heeft daarentegen wel iets nieuws gevonden. Het blijkt dat er soms meerdere kortste routes zijn. Deze routes zijn dan uiteraard allemaal even lang, maar verschillen op één of meerdere punten van elkaar.

Een spinnenweb bestaat hier altijd uit drie of meer spandraden, die allemaal vanuit het midden van het web uiteen lopen en één of meer tussendraden. De tussendraden zijn allemaal gesloten krommen en raken elkaar nooit. Zowel de spin als de vlieg bevinden zich altijd op een knooppunt van twee of meer draden. Een vlieg is niet helemaal gek en vliegt dus nooit het web in op de plaats waar de spin zit. Zodra de vlieg gevangen is in het web kan hij niet meer bewegen. De spin kan over alle draden en knooppunten lopen. Een stap is voor de spin een beweging over één van de draden vanaf het knooppunt waar hij zich bevindt tot aan het eerste knooppunt dat hij tegenkomt.



Probleem

Schrijf een programma dat, gegeven een beschrijving van een spinnenweb en de posities van de spin en de gevangen vlieg daarin, berekent langs hoeveel verschillende routes de spin, over zijn web, naar de vlieg kan kruipen, zodat hij daarbij een minimaal aantal stappen zet.

Invoer

De eerste regel van de invoer bevat een enkel positief getal: het aantal runs. Voor iedere run volgen er drie regels, met elk twee gehele getallen gescheiden door een spatie:

- Het aantal spandraden s en tussendraden t ($2 < s \leq 15$; $0 < t \leq 20$).
- De positie van de spin a, b ($0 < a \leq s$; $0 \leq b \leq t$).
- De positie van de vlieg c, d ($0 < c \leq s$; $0 \leq d \leq t$).

De tussendraden zijn genummerd van 1 tot t . Tussendraad 1 is de binnenste tussendraad en nummer t is de buitenste. Een positie met $b = 0$ of $d = 0$ betekent dat de spin, dan wel de vlieg, zich in het middelpunt van het web bevindt.

Uitvoer

Per run, steeds op een nieuwe regel, het aantal verschillende kortste routes.

Voorbeeld

invoer	bijbehorende uitvoer
2	3
3 3	16
2 1	
1 3	
10 5	
7 2	
1 4	

B Palindromië

Er was eens een land genaamd Palindromië, waar alle mensen dol waren op palindromen. Een *palindroom* is een woord waarbij de volgorde van de letters van links naar rechts exact hetzelfde is als van rechts naar links. 'parterretrap' is een heel bekend voorbeeld.

In een groot kasteel in Palindromië woont prinses Anna. Ze is heel mooi en heel lief, en prins Otto is dan ook waanzinnig verliefd op haar. Voordat hij met haar mag trouwen, moet hij echter het raadsel van de koning oplossen.

De koning heeft een lange reeks letters gekozen. Het is mogelijk om een palindroom te vormen door uit deze reeks bepaalde letters weg te laten; de overgebleven letters moeten daarbij in de oorspronkelijke volgorde blijven staan. De koning is op zoek naar het langste palindroom dat op deze manier gevormd kan worden.

Otto wil graag weten hoeveel letters hij weg moet laten om het langste palindroom te kunnen vinden. Met jouw hulp kunnen Otto en Anna trouwen en nog lang en gelukkig samen leven ...

Probleem

Gegeven een string van letters, hoeveel letters moet je minimaal uit de string weglaten om een palindroom over te houden.

Invoer

De invoer bevat een aantal afzonderlijke gevallen van het raadsel. Op de eerste regel staat het aantal gevallen n . Dan volgen n regels, met op elke regel een string. Elke string bevat uitsluitend kleine letters, en is minimaal 1 en maximaal 200 letters lang.

Uitvoer

Voor elk geval geef je 1 regel uitvoer met daarop het kleinste aantal letters dat moet worden weggelaten om een palindroom te vormen.

Voorbeeld

invoer	bijbehorende uitvoer
3	0
anna	3
hallo	14
programmeerwedstrijd	

C Boeken – weet je nog?

Guy Montag is een aardige jongen, maar hij houdt er wat vreemde ideeën op na. Hij denkt dat er informatie bestaat, belangrijke informatie, die niet op het Internet is te vinden. Ik bedoel: hij koopt soms een boek! Als hij een boek uit heeft legt hij het op een stapel. Als die stapel dreigt om te vallen begint hij een nieuwe stapel.

Dat wil zeggen, dat deed hij, tot hij Mildred tegenkwam. Mildred ontdekte bij een antiquair een paar enige boekenkastjes, en haalde Guy over die te kopen, en zijn boeken er in te zetten: “Ze passen er toch wel in?” Passen ze er in?

Probleem

Schrijf een programma dat nagaat of de boeken van Guy passen in de kasten van Mildred. Als ze niet passen moet het programma vertellen hoeveel er overblijven.

Een boek heeft een hoogte, bijvoorbeeld 200 mm, en een diepte. Als het boek gesloten voor je op tafel ligt zou je dat waarschijnlijk niet de diepte van het boek noemen, maar de breedte. De breedte van het boek bepaalt echter de diepte die een kast moet hebben, opdat het boek in die kast past. Daarom spreken we van de diepte van het boek.

Als regel zijn sommige boeken dikker dan andere. Of je het gelooft of niet, Guy bezit alleen maar boeken van 20mm dik!

Een boekenkast heeft een breedte, dat is de breedte aan de binnenkant. Deze breedte bepaalt het aantal boeken dat je naast elkaar op een plank in de kast kunt zetten.

Een boekenkast heeft een diepte. De diepte van de kast bepaalt de diepte van de boeken die je er in kunt zetten. Als de diepte van een kast 200 mm bedraagt, kun je boeken met een diepte van ten hoogste 200 mm in de kast plaatsen – tenminste als de hoogte dat toelaat.

Een boekenkast heeft een aantal planken. Elke plank heeft een hoogte. Eigenlijk is de hoogte niet een eigenschap van de plank, maar van de ruimte boven de plank. Op een plank met hoogte 280 mm kun je boeken kwijt met een hoogte van ten hoogste 280 mm – tenminste als de diepte dat toelaat.

Boeken worden rechtop in de kast gezet. Mildred heeft aan schief hangende boeken al net zo'n hekel als aan boeken op stapels.

Invoer

De eerste regel van de invoerfile bevat een enkel getal: het aantal testgevallen. De eerste regel van elk testgeval bevat een getal k ($0 < k < 10$), het aantal boekenkasten. Hierna volgen de k beschrijvingen van de boekenkasten.

De eerste regel van de beschrijving van een boekenkast bevat drie positieve getallen: de breedte, de diepte en het aantal planken p ($0 < p < 10$). De volgende regel bevat p getallen: de hoogte van de planken.

De beschrijving van de kasten wordt gevolgd door de beschrijving van de boeken. De eerste regel van de beschrijving van de boeken bevat een positief getal b ($b < 1000$), het aantal boeken.

De volgende b regels bevatten elk twee getallen: eerst de diepte, dan de hoogte van een boek.

Alle afmetingen zijn gehele getallen in millimeters. Breedten, diepten en hoogten zijn ten hoogste 1000 mm.

Uitvoer

De uitvoer bevat, voor elk testgeval op een aparte regel, het aantal boeken dat overblijft, nadat er zo veel mogelijk boeken in de kasten zijn gezet. Als alle boeken in de kasten passen is dat aantal natuurlijk 0.

Voorbeeld

invoer	bijbehorende uitvoer
2	1
2	0
1000 200 3	
180 210 250	
700 250 2	
300 300	
3	
250 300	
200 280	
300 200	
1	
22 200 1	
250	
1	
200 250	

D Schrikkeljaren

Iedere programmeur heeft of krijgt nog wel eens te maken met het fenomeen *schrikkeljaar*¹. De oorzaak van een schrikkeljaar is gelegen in de duur van de omloop van de aarde rond de zon, gemeten in dagen. Dit is helaas niet precies een geheel getal en daarom is het nodig om eens in de vier jaar een extra dag in een jaar te stoppen.

Dit is weer iets teveel zodat er in eeuwjaren (jaartal deelbaar door honderd) geen schrikkeljaar is – althans, vanaf 1800; in 1700 zag men het probleem nog niet zo en dit was dus gewoon een schrikkeljaar. Op de “eeuwjaren”-regel is er dan nog de uitzondering dat eeuwjaren, deelbaar door 400, juist weer wel schrikkeljaar zijn! Deze laatste regel werd vorig jaar (2000) voor het eerst toegepast.

Bovenstaande regels zijn specifiek geschikt voor de aarde. Op andere planeten zijn andere regelingen, afhankelijk van de precieze omlooptijd (gemeten in dagen) een stuk handiger.

Het is 2001, en je bent op weg naar Saturnus in het ruimteschip Endeavour. De verwachting is dat deze reis zal leiden tot contact met buitenaardse beschavingen, elk met een vreemde schrikkeljaar-regeling.

Probleem

Aan jou de vraag om een programma te schrijven waarmee de boordcomputer (een HRN-9000) op basis van een gespecificeerde schrikkeljaar-regeling, de omlooptijd in dagen weer kan geven.

Invoer

Op de eerste regel staat het aantal runs R . Hierna volgen precies R schrikkeljaar-regeling specificaties. Elke specificatie begint met het aantal regels N ($1 \leq N \leq 20$) waaruit de regeling bestaat. Hierna volgen N regels met twee getallen, D_i en L_i ($1 \leq D_i \leq 10000$, $1 \leq L_i \leq 10000$, $1 \leq i \leq N$), die een deler D_i aangeven met een bijbehorende jaar-lengte L_i .

Zo'n regel moet als volgt worden gelezen: “indien het jaartal deelbaar is door D_i , dan is de lengte van het jaar gelijk aan L_i dagen”.

In de schrikkeljaar-regelingen zoals hier beschouwd geldt altijd dat de eerst passende regel wordt gebruikt om te kijken of een jaar schrikkeljaar is; indien een “match” wordt gevonden, worden de overige regels genegeerd.

¹Een echte programmeur zorgt er immers voor dat zijn of haar programma in ieder geval tot het jaar 2038 correct blijft werken, en liefst nog veel langer!

Regelingen die op deze wijze worden gespecificeerd zijn altijd periodiek (i.e., herhalen zich na een aantal jaren); er is gegeven dat deze periode een lengte heeft van ten hoogste 10000 jaar. Verder is gegeven dat er altijd een regel is met $D_i = 1$, zodat elk jaar een goed gedefinieerde lengte heeft.

NB. Sommige schrikkeljarenstelsels bevatten overbodige, of zelfs strijdige regels! Zoals gezegd wordt in alle gevallen altijd de eerste regel die opgaat toegepast.

Uitvoer

R regels, met op elke regel de berekende omlooptijd voor een schrikkeljaar-regeling, uitgedrukt in dagen. Druk het getal af met altijd precies vijf cijfers achter de decimale punt. Rond af op de gebruikelijke manier: omhoog als het volgende cijfer ≥ 5 , anders omlaag.

Voorbeeld

invoer	bijbehorende uitvoer
2	365.24250
4	115.08000
400 366	
100 365	
4 366	
1 365	
3	
25 117	
50 119	
1 115	

E Binaire Codes

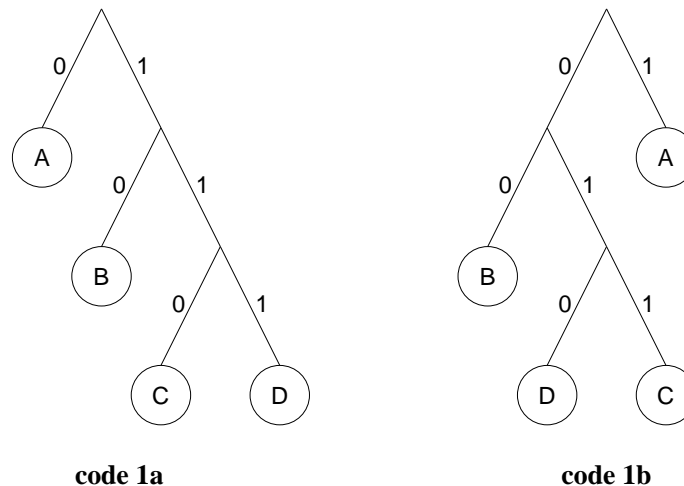
Een vaak gebruikte manier om data te comprimeren is om de te coderen symbolen te vervangen door bitrijen van verschillende lengte. Stel dat we bijvoorbeeld een bestand hebben met de 4 symbolen A, B, C en D, die in de volgende verhouding voorkomen:

symbool	percentage
A	54%
B	42%
C	3%
D	1%

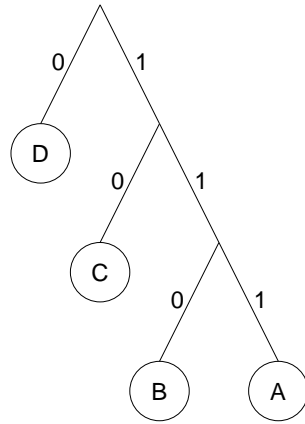
In dit geval is het voordelig om de A te representeren door 1 bit, de B door 2 bits, en de C en D door 3 bits. Dit kan op verschillende manieren. Hieronder zijn er twee gegeven:

symbool	code 1a	code 1b
A	0	1
B	10	00
C	110	011
D	111	010

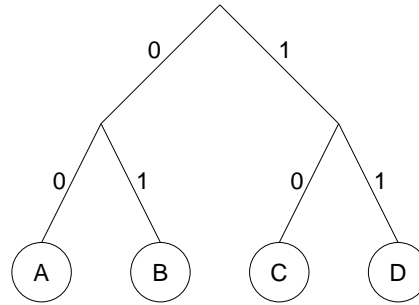
Merk op dat we ervoor zorgen dat ‘terugvertalen’ eenvoudig is, door de codewoorden zo te kiezen dat langere codewoorden nooit beginnen met een bitrij die ook een korter codewoord is. Dergelijke codes worden *prefix-codes* genoemd. Prefix-codes kunnen handig worden weergegeven in een binaire boom, waarin iedere interne knoop overeenkomt met de keuze tussen 0 en 1, en de bladen van de boom overeenkomen met een gecodeerd symbool:



Bovenstaande twee prefix-codes zijn *equivalent* in de zin dat ze, ongeacht het te comprimeren bestand, altijd dezelfde graad van compressie zullen opleveren. Er bestaan echter ook niet-equivalente prefix-codes, die andere compressie-eigenschappen hebben, bijvoorbeeld:



code 2



code 3

De boom voor code 2 lijkt weliswaar op die voor code 1a, maar er zijn symbolen verwisseld waardoor de compressie-eigenschappen van de code heel anders zijn. Van code 3 is meteen duidelijk dat het een heel andere code is.

Een handige manier om alle niet-equivalente prefix-codes te onderscheiden is door een lijst te maken waarin voor ieder symbool is gegeven door hoeveel bits het gerepresenteerd wordt. Voor vier symbolen blijken er dan 13 verschillende niet-equivalente prefix-codes te bestaan:

symbool	bits per symbool
A	1 1 1 2 3 3 2 3 3 2 3 3 2
B	2 3 3 1 1 1 3 2 3 3 2 3 2
C	3 2 3 3 2 3 1 1 1 3 3 2 2
D	3 3 2 3 3 2 3 3 2 1 1 1 2

Probleem

Gegeven een aantal symbolen, hoeveel niet-equivalente prefix-codes zijn er dan mogelijk?

Invoer

De eerste regel bevat een aantal runs R . Op de R volgende regels staat steeds een getal S , dat het aantal te coderen symbolen bevat ($2 \leq S \leq 100$).

Uitvoer

R regels met het aantal mogelijke niet-equivalente prefix-codes. Gezien de grootte van de getallen dienen de aantallen te worden weergegeven als 2^x , waarbij x een op drie cijfers achter de decimale punt afgerond getal is. Rond af op de gebruikelijke manier: omhoog als het volgende cijfer ≥ 5 , anders omlaag.

Voorbeeld

invoer	bijbehorende uitvoer
5	$2^{0.000}$
2	$2^{1.585}$
3	$2^{3.700}$
4	$2^{6.229}$
5	$2^{548.176}$
100	

F Verkiezingen

Het koninkrijk der Verenigde Steden wordt bewoond door twee stammen: de Burgers en de Nomaden. De Burgers wonen in de steden. De Nomaden daarentegen hebben geen vaste woon- of verblijfplaats, zij trekken rond met hun vee.

Elke vijf jaar wordt in de Verenigde Steden een nieuwe koning gekozen. Deze verkiezing vindt in twee fasen plaats. In de eerste fase kiest elke stad zijn kiesmannen, in de tweede fase kiezen alle kiesmannen samen in een gezamenlijke slotvergadering de nieuwe koning.

In de eerste fase wordt in elke stad bepaald welke kiesmannen naar de slotvergadering gaan. De kiezers stemmen in deze fase echter niet op een kiesman, zoals men zou verwachten, maar op een van de konings-kandidaten. Indien één van de konings-kandidaten in een stad de meerderheid van de stemmen heeft, dan zullen alle kiesmannen van deze stad in de slotvergadering voor deze kandidaat kiezen (als er totaal s stemmen worden uitgebracht, dan is m een meerderheid als $2 \times m > s$). Als geen van de kandidaten de meerderheid heeft in een stad, dan gaan er geen kiesmannen van deze stad naar de slotvergadering. Vroeger gebeurde dat nog wel eens, als er meer dan twee kandidaten waren. Tegenwoordig zijn er eigenlijk altijd maar twee kandidaten: de kandidaat van de Burgers, en de kandidaat van de Nomaden. Nochtans kunnen de stemmen staken, zodat er geen kiesmannen naar de slotvergadering gaan.

Het aantal kiesmannen dat een stad mag afvaardigen naar de slotvergadering (de tweede fase) hangt af van het totaal aantal uitgebrachte stemmen in de stad: voor elke volle duizend uitgebrachte stemmen mag de stad één kiesman afvaardigen. (Sommige mensen gaan om die reden niet stemmen. Als je weet dat jouw kandidaat toch niet gaat winnen, draagt jouw stem alleen maar bij aan het aantal kiesmannen voor de andere kandidaat.)

In de tweede fase, de slotvergadering, wordt de koning gekozen door de gezamenlijke kiesmannen. Een koning is gekozen als hij of zij de meerderheid van de stemmen heeft in de slotvergadering. Er zijn regels die voorschrijven wat te doen als geen van de kandidaten een meerderheid krijgt, maar die doen voor deze opgave niet terzake.

Hoewel er veel meer Nomaden dan Burgers in de Verenigde Steden wonen, werd in het verleden meestal de kandidaat van de Burgers gekozen. De Nomaden moesten naar de steden gaan om te stemmen, maar menig Nomade had niet eens in de gaten dat er verkiezingen waren. Tegenwoordig hebben de meeste Nomaden mobiele telefoons. Het verkiezingsteam van de Nomaden maakt daar dankbaar gebruik van, en dirigeert zoveel Nomaden naar de steden als nodig is om de verkiezingen te winnen. Van elke stad is bekend hoeveel Burgers er wonen, en we mogen er van uitgaan dat die voor hun Burger kandidaat stemmen.

Probleem

Neem aan dat de Nomaden zo slim mogelijk naar de steden worden gedirigeerd. Wat is het minimum aantal Nomaden dat moet gaan stemmen? De bedoeling is natuurlijk dat de kandidaat van de Nomaden wint.

Invoer

De eerste regel van de invoerfile bevat een positief getal: het aantal testgevallen. De eerste regel van elk testgeval bevat het aantal steden n ($0 < n < 20$). De volgende n regels bevatten ieder één getal: het aantal (Burger-)bewoners van de stad. Dit aantal is altijd positief. Een stad heeft nooit meer dan 100 000 inwoners.

Uitvoer

De uitvoer bevat, voor elk testgeval in de invoerfile, op een aparte regel, het minimale aantal Nomaden dat moet gaan stemmen om te zorgen dat de kandidaat van de Nomaden wint.

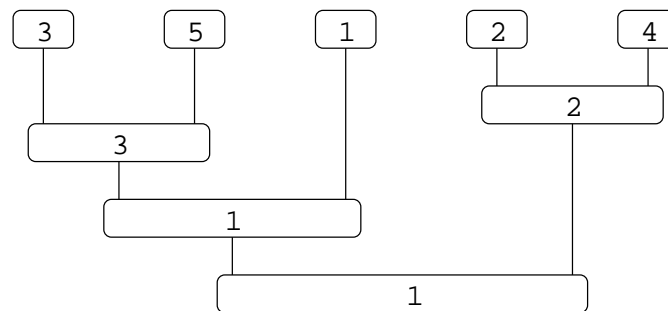
Voorbeeld

invoer	bijbehorende uitvoer
2	1200
3	1001
400	
800	
1200	
3	
999	
1000	
1001	

G De Fusiegolf

De immer alerte Argus van de Nieuwe Rommeldamsche Courant wil in een alarmerend artikel het ragfijn spel van het GROOTKAPITAAL in de om zich heen grijpende fusiegolf aan de kaak stellen. Hij beschikt over een lijst van de omzetten van de companies in miljoenen dollars (sorry boys, ze kumpeni lengwidz is inglisch), en een chronologische lijst van fusies van telkens twee companies die samen gaan onder de naam van de oudste.

Argus vreest niet ten onrechte dat zijn lezertjes het betoog niet zullen kunnen volgen zonder een schematisch diagram van het verloop van de fusies. Kruisende lijnen zouden het diagram onleesbaar maken, daarom wil hij de companies zo ordenen dat steeds twee burenen fuseren, en bovendien wil hij – om zijn punt te maken – de grootste company steeds voorop hebben. Op de journalistenschool heeft hij zo iets nooit geleerd. Help hem!



Probleem

Gegeven zijn een aantal companies, genummerd van 1 tot en met NC , waarbij de laagste nummers corresponderen met de oudste bedrijven, de omzetten van de bedrijven en een chronologische lijst van fusies. Bepaal de volgorde waarin Argus de bedrijven moet ordenen om een diagram te krijgen dat aan bovenstaande eisen voldoet.

Invoer

De input bestaan uit N gevallen. De eerste regel van de input bevat alleen het positieve getal N . Daarna volgen de afzonderlijke gevallen. De eerste regel van een geval bevat een geheel getal NC : het aantal companies ($1 \leq NC \leq 4000$). Hierna volgen NC regels met de omzetten van de companies ($0 \leq \text{omzet} \leq 10000$). Vervolgens $NC - 1$ regels met telkens 2 getallen gescheiden door een spatie: de fusiepartners van een fusie.

De invoer is zodanig dat de gevraagde volgorde eenduidig bepaald is.

Uitvoer

Voor ieder van de gevallen een enkele regel met daarop een opsomming van de nummers van de companies in de gevraagde volgorde. Nummers van companies dienen te worden gescheiden door een spatie.

Voorbeeld

invoer	bijbehorende uitvoer
1	3 5 1 2 4
5	
100	
110	
90	
40	
80	
2 4	
3 5	
1 3	
1 2	

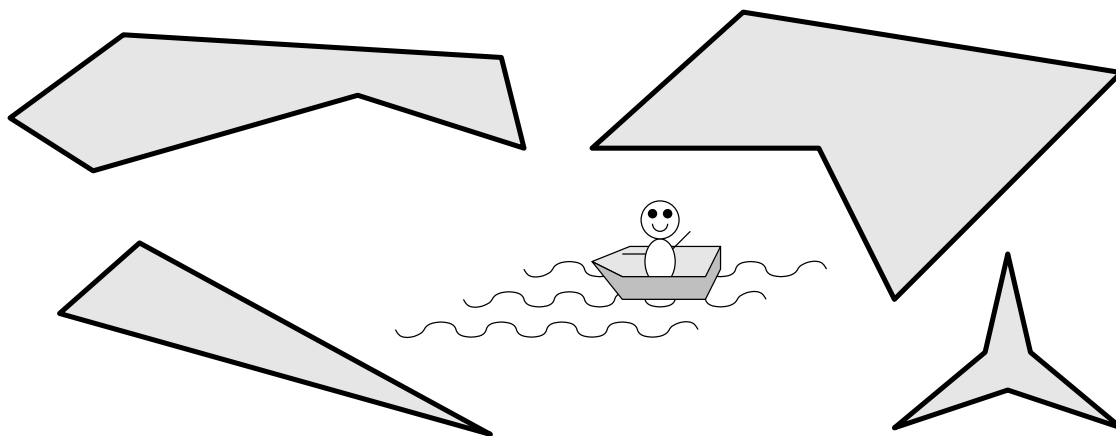
H De stranden van Polygonesië

Zoals bekend is *Polygonesië* een eilandengroep in de Verre Oceaan. Elk eiland heeft de vorm van een polygon (veelhoek); dat wil zeggen: de kust wordt gevormd door een verzameling aaneengesloten rechte lijnen.

De Polygonesische eilanden zijn nogal klein en liggen vrij dicht bij elkaar. Als je met een bootje in de buurt van de eilanden op zee vaart, kun je ze allemaal zien liggen. Dat wil zeggen, voor zover ze niet achter andere eilanden verscholen liggen.

De Kleine Kapitein maakt een lange reis met zijn roeiboot. Onderweg vaart hij toevallig door Polygonesië. Wanneer hij om zich heen naar de horizon kijkt, ziet hij in sommige richtingen uitsluitend water en lucht. In andere richtingen kan hij echter in de verte een strand van een Polygonesisch eiland zien liggen.

De kapitein vraagt zich af welk deel van de horizon in beslag wordt genomen door eilandkust, en welk deel slechts uitzicht biedt op het water van de zee. Misschien kan hij met behulp van de boordcomputer van de roeiboot een antwoord vinden op dit belangrijke vraagstuk.



Probleem

Verplaats je in de situatie van de kapitein. Schrijf een programma dat voor een gegeven positie van de roeiboot, berekent welke fractie van de horizon door de eilanden wordt bezet.

Invoer

De invoer beschrijft een aantal afzonderlijke situaties. Op de eerste regel staat het aantal situaties in de vorm van een positief geheel getal. Dan volgt voor elke situatie:

- Een regel met het aantal eilanden ($1 \leq n \leq 20$).
- Vervolgens n regels met op elke regel de beschrijving van een eiland. Een eiland wordt beschreven door een opsomming van gehele getallen, van elkaar gescheiden door spaties. Het eerste getal is het aantal hoekpunten p van het eiland ($3 \leq p \leq 20$). Daarachter komen nog p maal twee getallen: de x - en y -coördinaten van de hoekpunten in meters ($0 \leq x, y \leq 1000$). Tussen twee opeenvolgende hoekpunten **en ook** tussen het laatste en het eerste hoekpunt van een eiland, loopt de eilandkust in een rechte lijn.
- Tot slot een regel met twee gehele getallen, gescheiden door een spatie: de x - en y -coördinaten van de roeiboot ($0 \leq x, y \leq 1000$). De boot ligt vrij in het water, los van alle eilanden.

Opmerking: Elk eiland vormt een aaneengesloten gebied. Alle eilanden liggen los van elkaar.

Uitvoer

Voor elke situatie geef je één regel, met daarop het deel van de horizon waar eilandkust te zien is, in de vorm van een getal k ($0 \leq k \leq 1$). Schrijf het antwoord altijd met precies 1 cijfer voor de decimale **punt** en precies 4 cijfers er achter. Rond af op de gebruikelijke manier: omhoog als volgende cijfer ≥ 5 , anders omlaag.

Voorbeeld

invoer	bijbehorende uitvoer
2	0.2500
1	0.5738
3 0 0 0 1 1 0	
1 1	
2	
4 0 0 0 3 1 1 3 0	
3 2 1 4 1 4 2	
2 2	